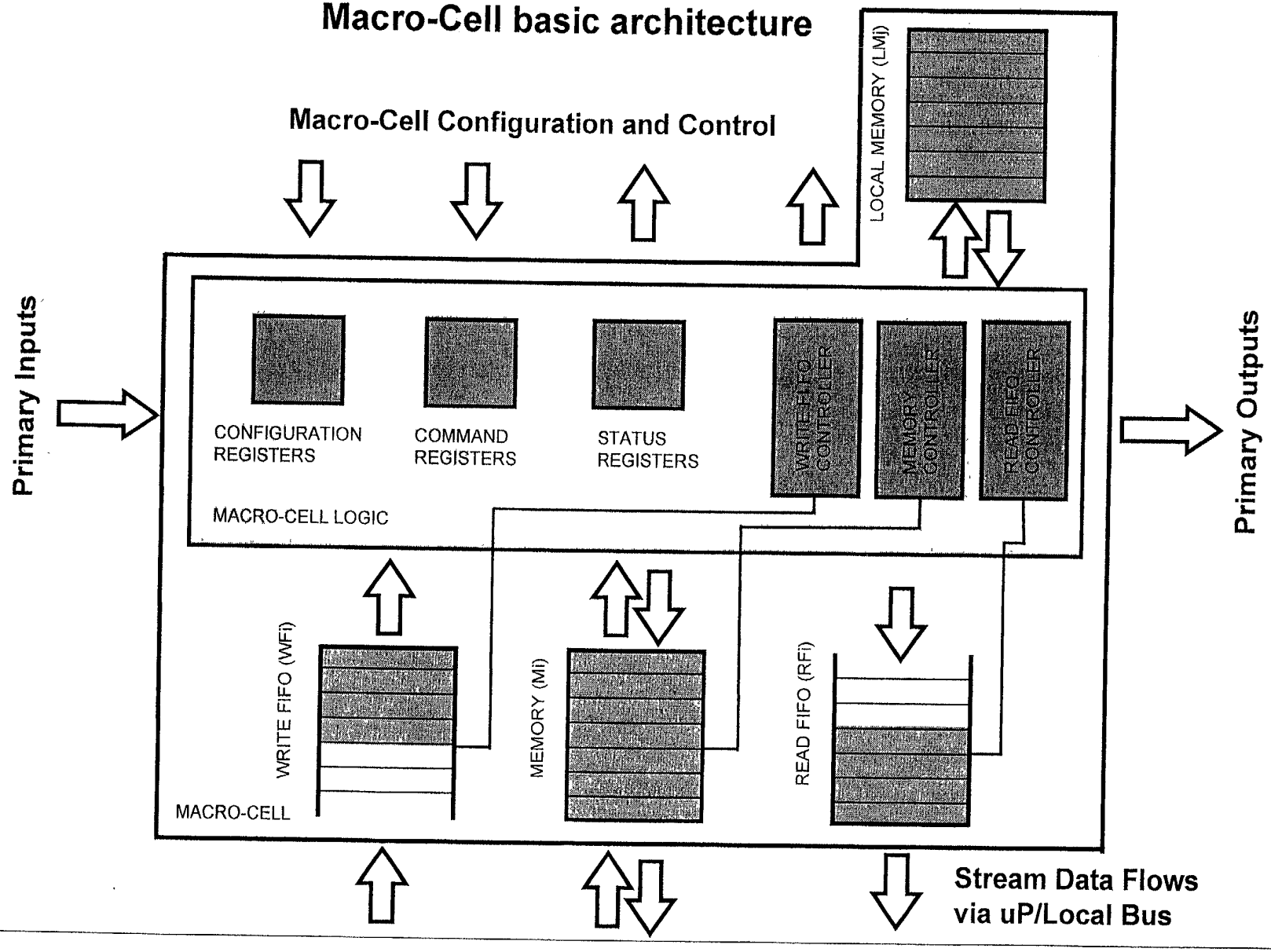
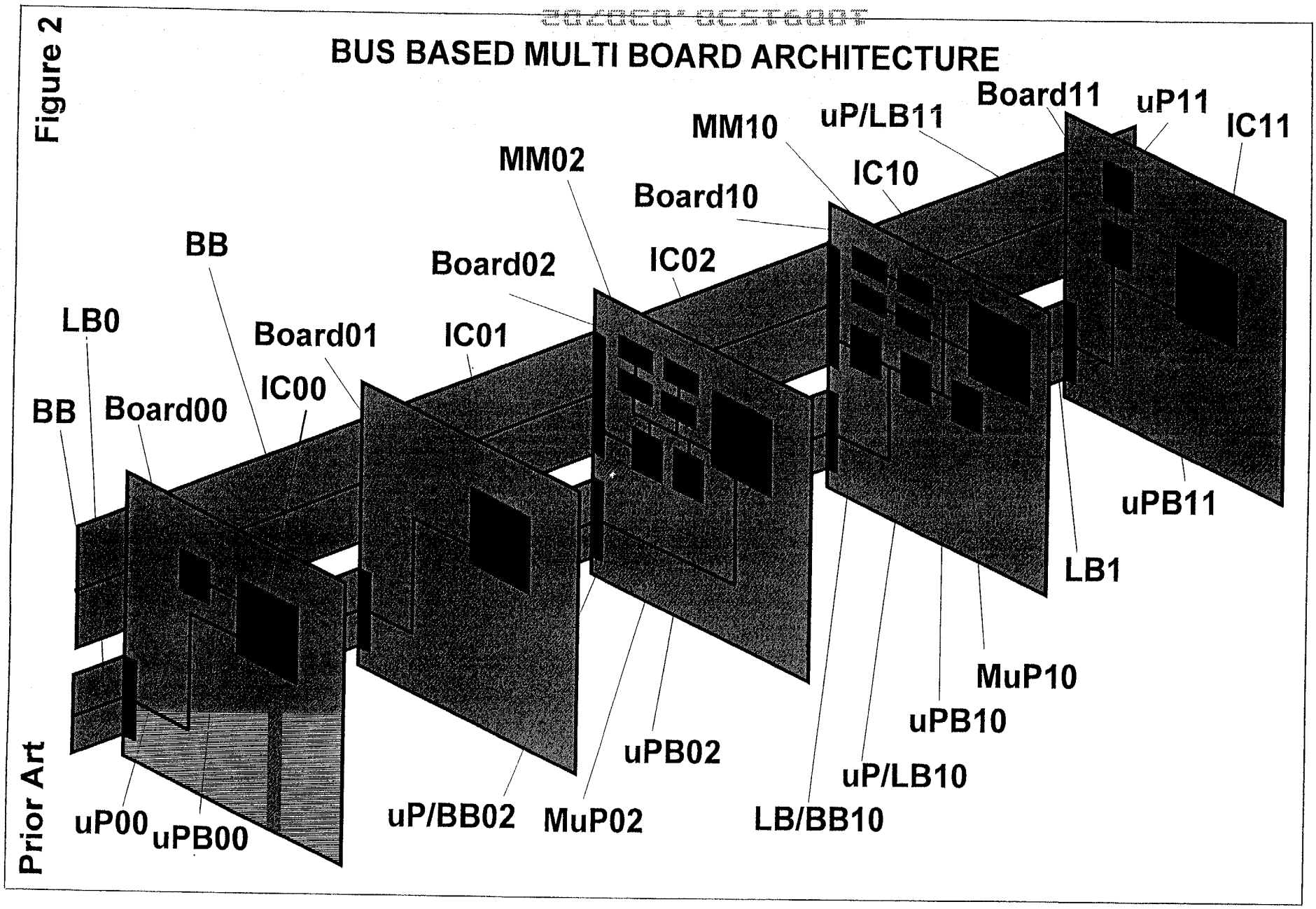
Figure 1



Prior Art

Figure 2

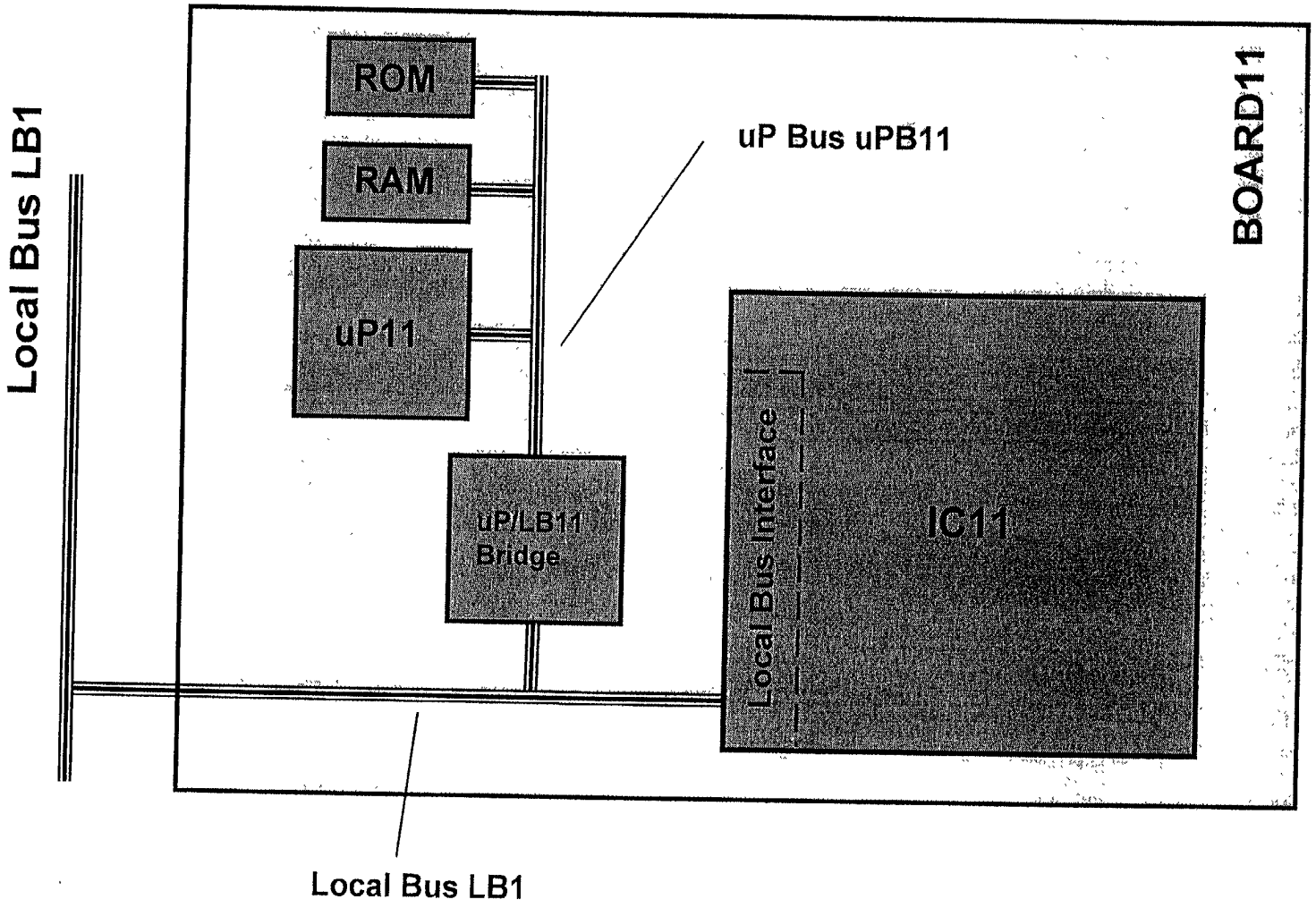

BUS BASED MULTI BOARD ARCHITECTURE
Board11
uP11
IC11
MM10
uP/LB11
IC10
MM02
Board10
IC02
BB
Board02
LB0
Board01
IC01
IC00
BB
Board00
uPB11
LB1
MuP10
uPB10
uP/LB10
uPB02
LB/BB10
uP/BB02
MuP02
uP00
uPB00
Prior Art

Prior Art

Figure 3

Figure 4

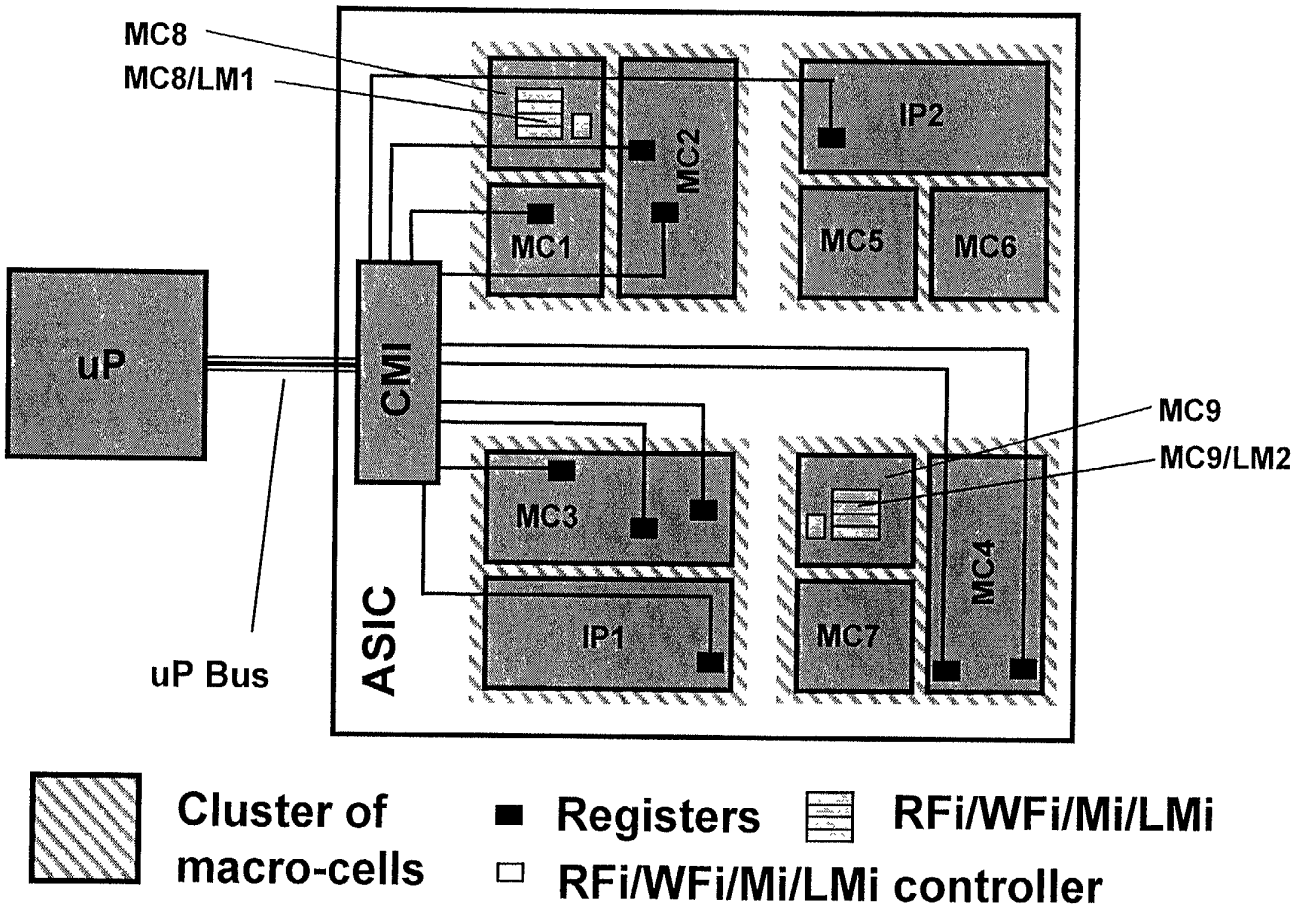
IC WITH LOCAL BUS INTERFACE



Prior Art

Figure 5

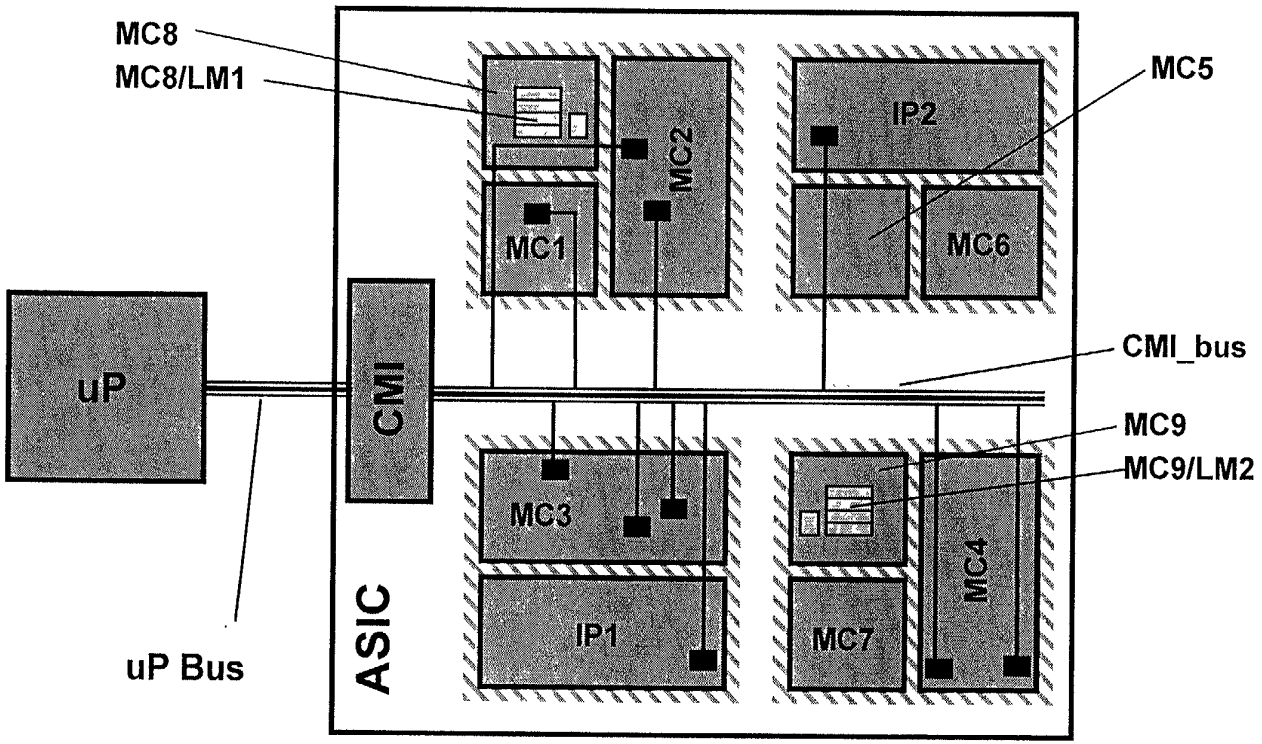
ASIC implementation of CMI with CMPI macro-cells interface



Prior Art

Figure 6

ASIC implementation of CMI with CBBI macro-cells interface



Cluster of macro-cells

Registers

RFi/WFi/Mi/LMi

RFi/WFi/Mi/LMi controller

Prior Art

Figure 7

Prior Art

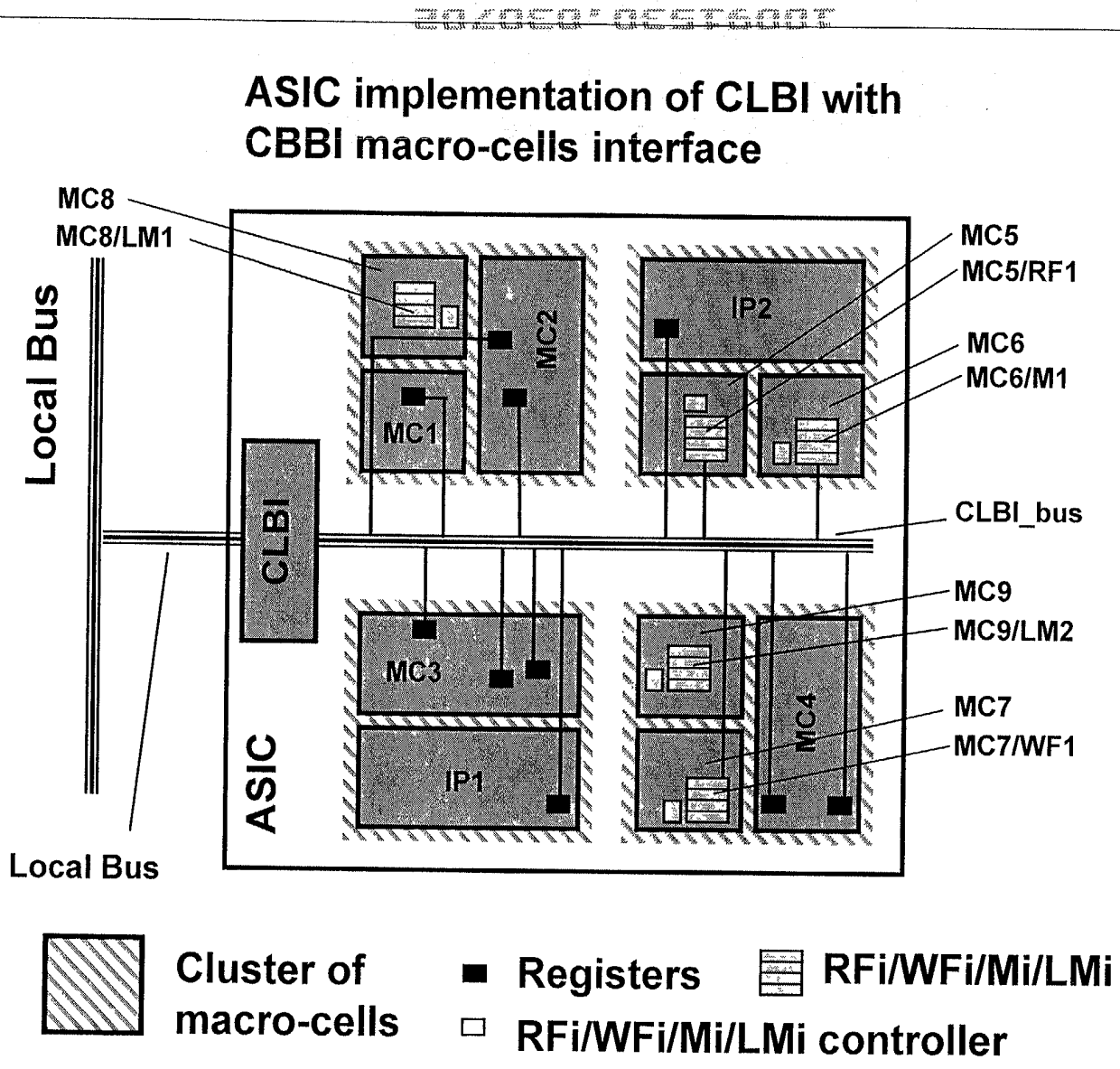
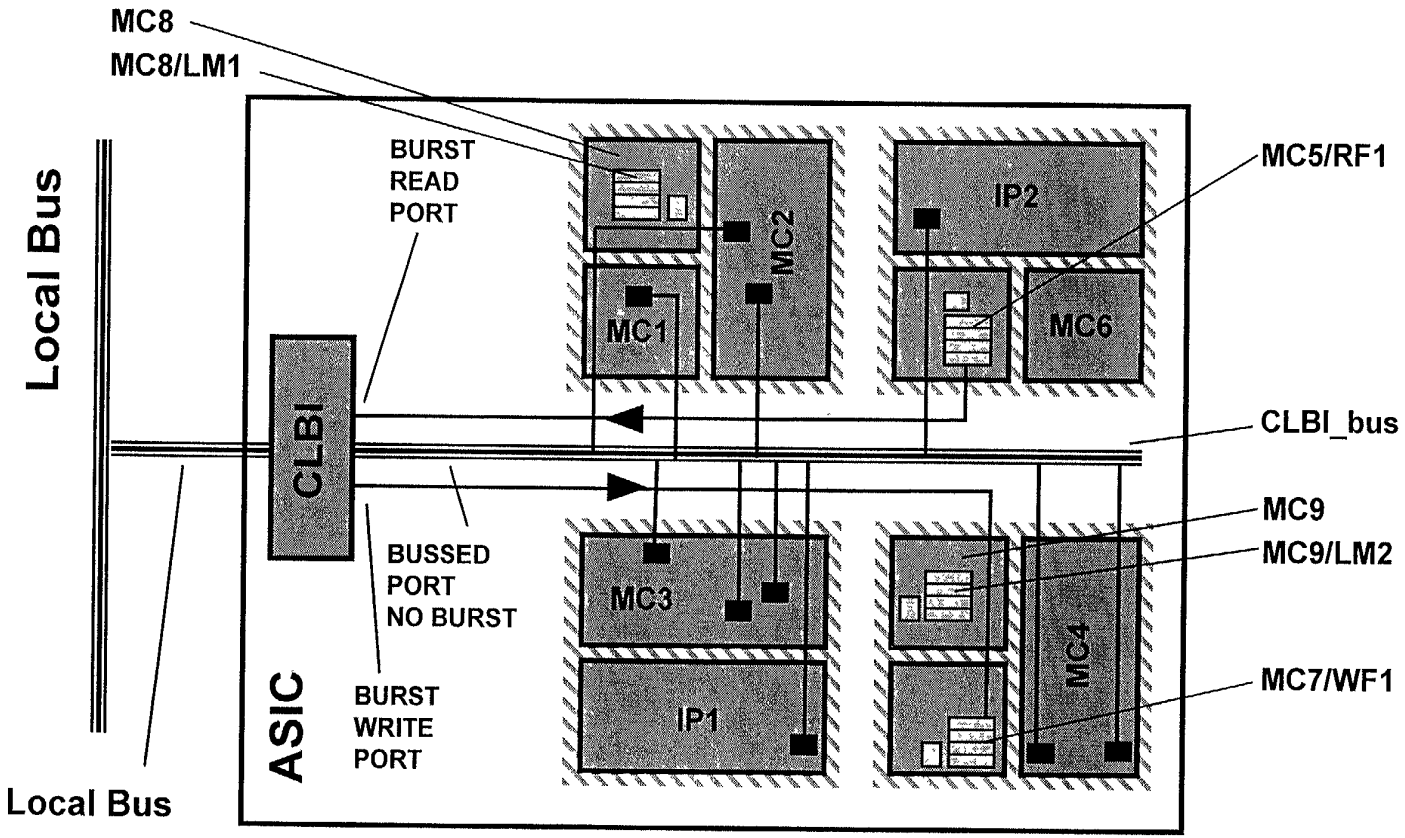


Figure 8

ASIC implementation of CLBI with CMPI macro-cells interface



Cluster of macro-cells

Registers

RFi/WFi/Mi/LMi

RFi/WFi/Mi/LMi controller

Prior Art

Figure 9

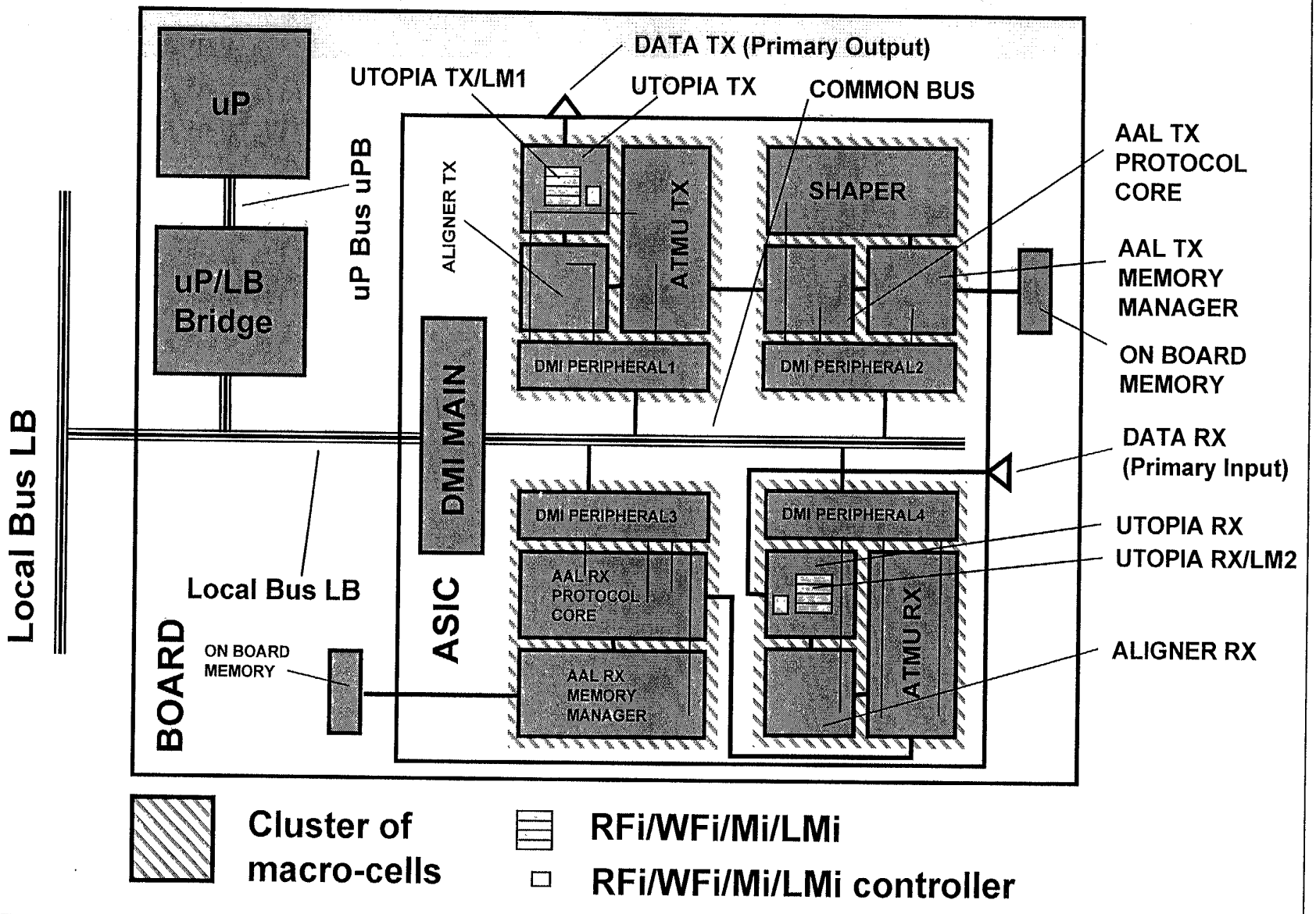
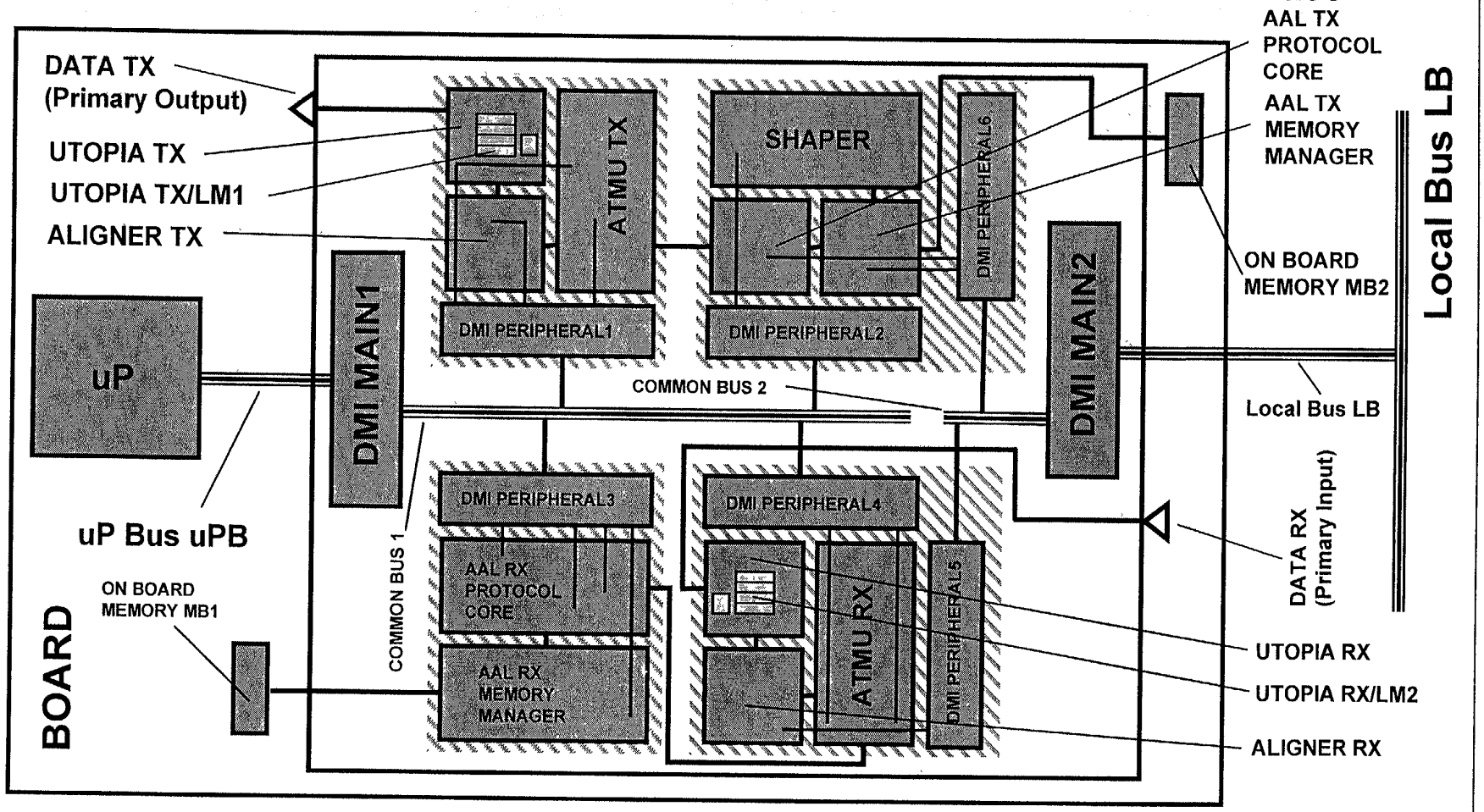

Board hosting ASIC implementation of AAL5 interfaced via DMI
uP
uP/LB Bridge
uP Bus uPB
Local Bus LB
BOARD
ON BOARD MEMORY
Local Bus LB
ASIC
DMI MAIN
ALIGNER TX
ATMU TX
DMI PERIPHERAL1
SHAPER
DMI PERIPHERAL2
DMI PERIPHERAL3
AAL RX PROTOCOL CORE
AAL RX MEMORY MANAGER
DMI PERIPHERAL4
ATMU RX
DATA TX (Primary Output)
UTOPIA TX/LM1
UTOPIA TX
COMMON BUS
AAL TX PROTOCOL CORE
AAL TX MEMORY MANAGER
ON BOARD MEMORY
DATA RX (Primary Input)
UTOPIA RX
UTOPIA RX/LM2
ALIGNER RX
Cluster of macro-cells
RFi/WFi/Mi/LMi
RFi/WFi/Mi/LMi controller

Figure 10

Board hosting FPGA bread-boarding implementation of a DMI for microprocessor interface and a DMI for local bus interface



Cluster of macro-cells

RFi/WFi/Mi/LMi

RFi/WFi/Mi/LMi controller

Figure 11

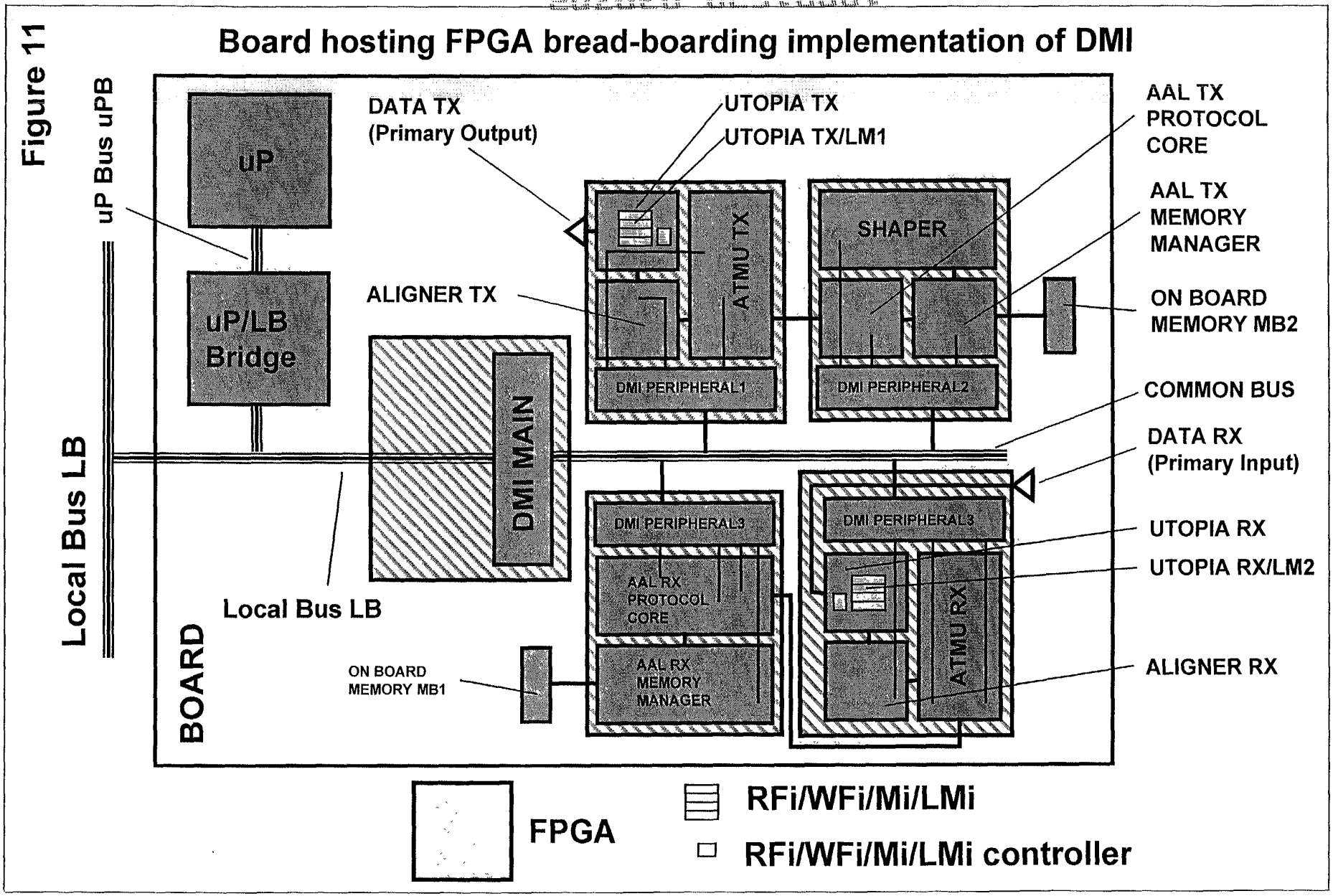


Figure 12

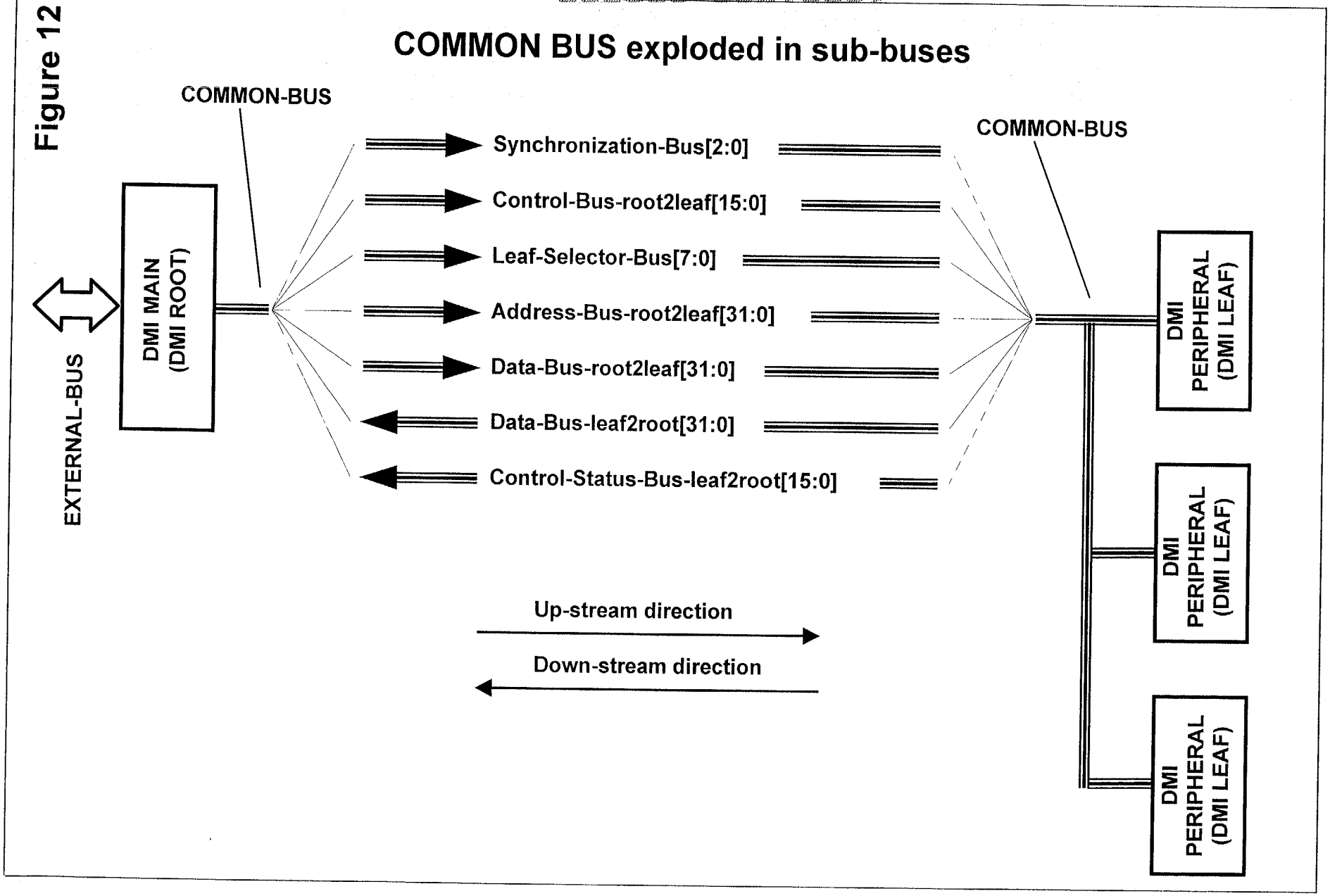

COMMON BUS exploded in sub-buses
COMMON-BUS
COMMON-BUS
EXTERNAL-BUS
DMI MAIN (DMI ROOT)
Synchronization-Bus[2:0]
Control-Bus-root2leaf[15:0]
Leaf-Selector-Bus[7:0]
Address-Bus-root2leaf[31:0]
Data-Bus-root2leaf[31:0]
Data-Bus-leaf2root[31:0]
Control-Status-Bus-leaf2root[15:0]
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)
Up-stream direction
Down-stream direction

Figure 13

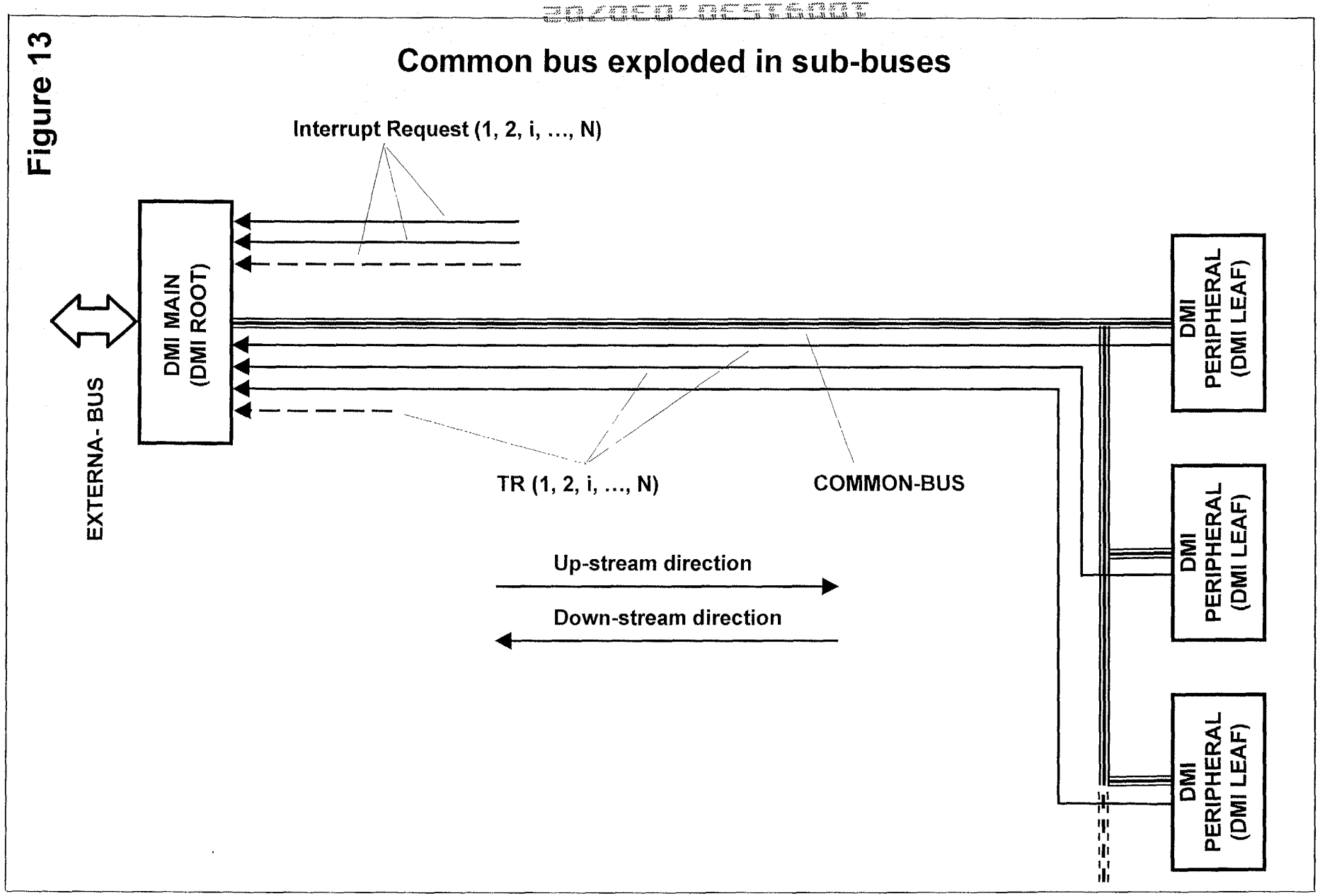

Common bus exploded in sub-buses
Interrupt Request (1, 2, i, ..., N)
DMI MAIN (DMI ROOT)
EXTERNA- BUS
TR (1, 2, i, ..., N)
COMMON-BUS
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)
DMI PERIPHERAL (DMI LEAF)
Up-stream direction
Down-stream direction

Figure 14

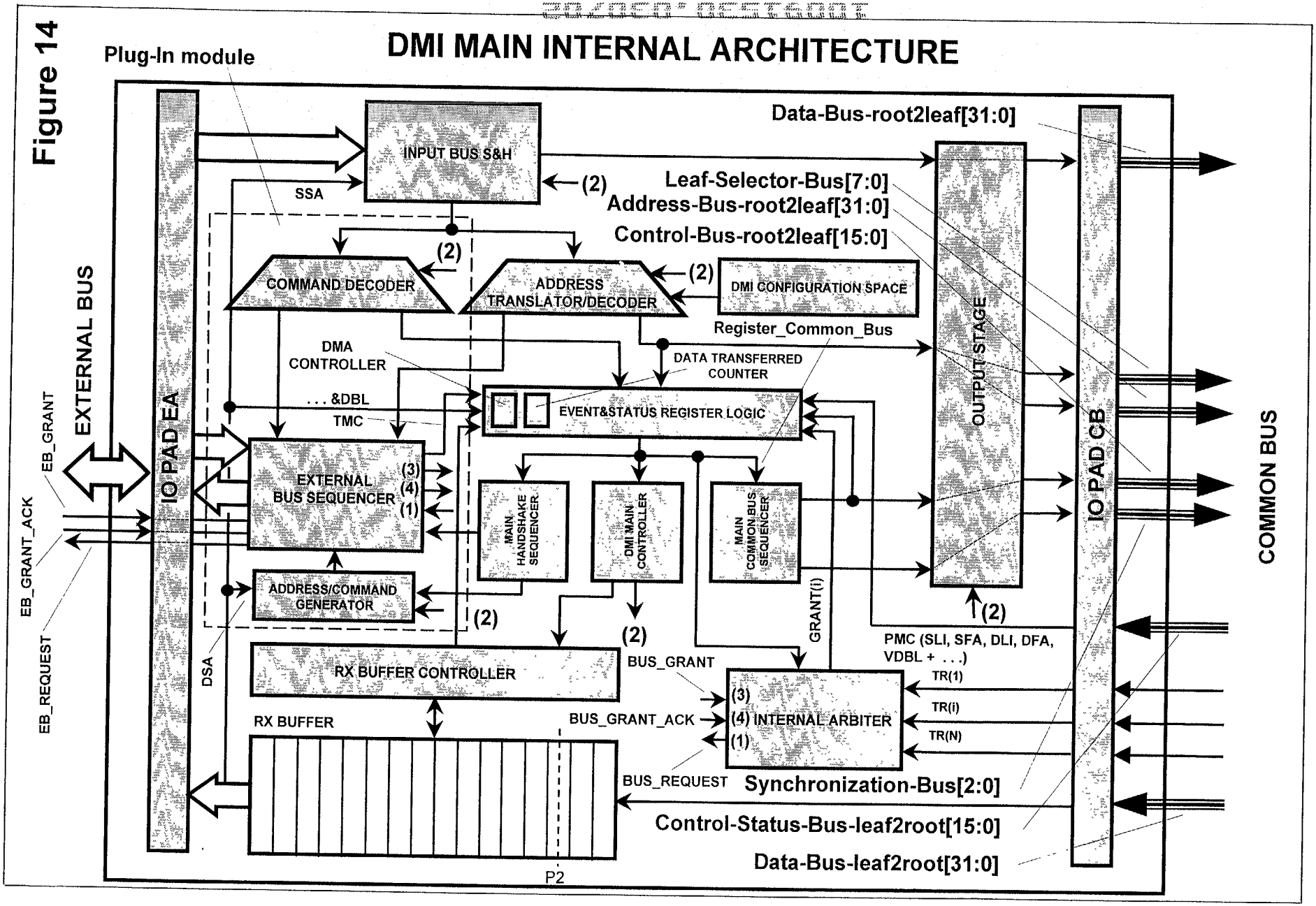

DMI MAIN INTERNAL ARCHITECTURE
Plug-In module
EXTERNAL BUS
EB_GRANT
EB_GRANT_ACK
EB_REQUEST
IO PAD EA
INPUT BUS S&H
SSA
(2)
Data-Bus-root2leaf[31:0]
Leaf-Selector-Bus[7:0]
Address-Bus-root2leaf[31:0]
Control-Bus-root2leaf[15:0]
COMMAND DECODER
ADDRESS TRANSLATOR/DECODER
DMI CONFIGURATION SPACE
Register_Common_Bus
DMA CONTROLLER
DATA TRANSFERRED COUNTER
. . . &DBL
TMC
EVENT&STATUS REGISTER LOGIC
EXTERNAL BUS SEQUENCER
(3)
(4)
(1)
MAIN HANDSHAKE SEQUENCER
DMI MAIN CONTROLLER
MAIN COMMON BUS SEQUENCER
OUTPUT STAGE
IO PAD CB
COMMON BUS
ADDRESS/COMMAND GENERATOR
DSA
GRANT(i)
PMC (SLI, SFA, DLI, DFA, VDBL + . . .)
RX BUFFER CONTROLLER
BUS_GRANT
TR(1)
TR(i)
TR(N)
RX BUFFER
BUS_GRANT_ACK
(4) INTERNAL ARBITER
BUS_REQUEST
Synchronization-Bus[2:0]
Control-Status-Bus-leaf2root[15:0]
Data-Bus-leaf2root[31:0]
P2

Figure 15

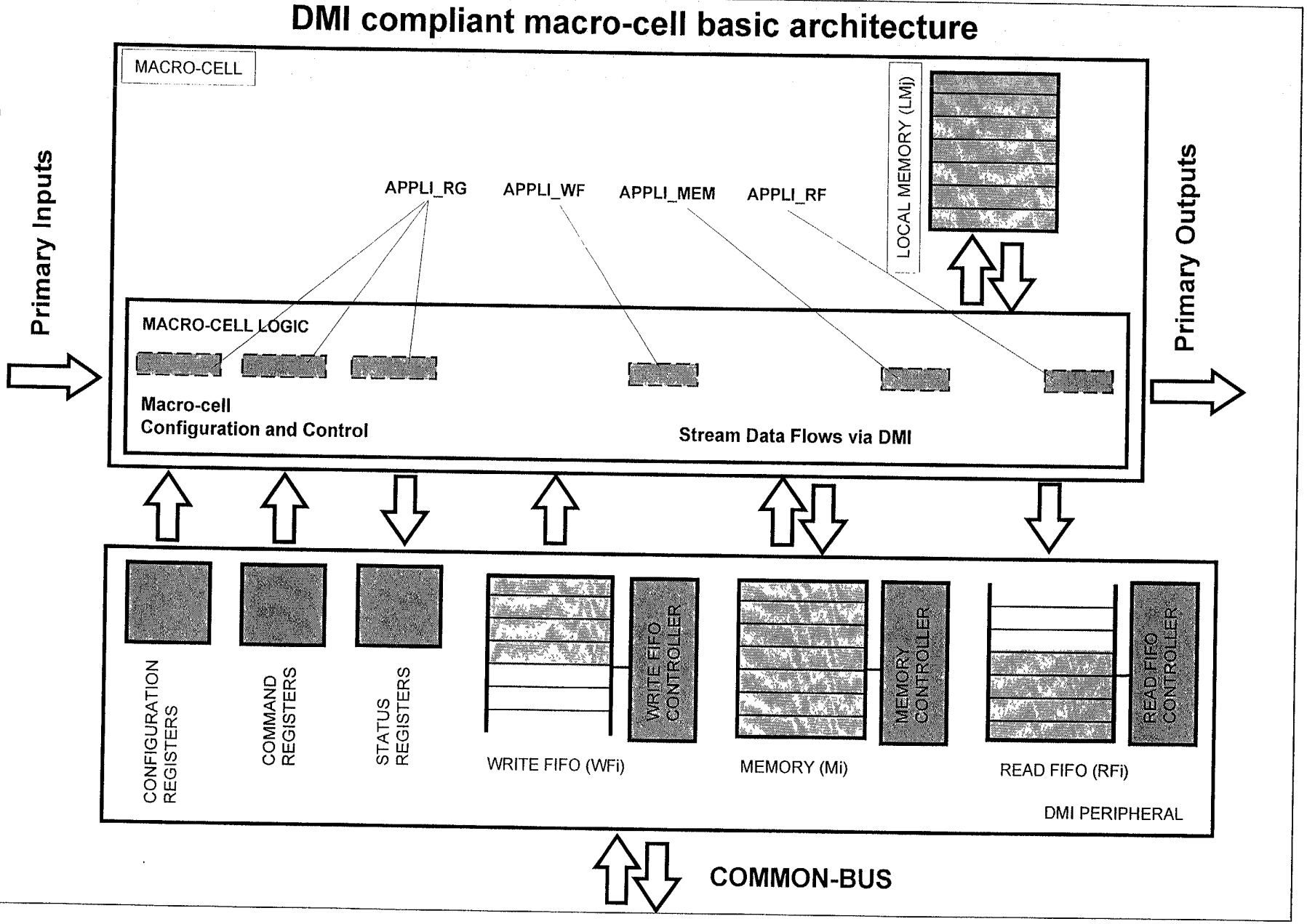

DMI compliant macro-cell basic architecture
MACRO-CELL
Primary Inputs
Primary Outputs
LOCAL MEMORY (LMi)
APPLI_RG
APPLI_WF
APPLI_MEM
APPLI_RF
MACRO-CELL LOGIC
Macro-cell
Configuration and Control
Stream Data Flows via DMI
CONFIGURATION REGISTERS
COMMAND REGISTERS
STATUS REGISTERS
WRITE FIFO CONTROLLER
MEMORY CONTROLLER
READ FIFO CONTROLLER
WRITE FIFO (WFi)
MEMORY (Mi)
READ FIFO (RFi)
DMI PERIPHERAL
COMMON-BUS

Figure 16

DMI PERIPHERAL Shadowed Layers

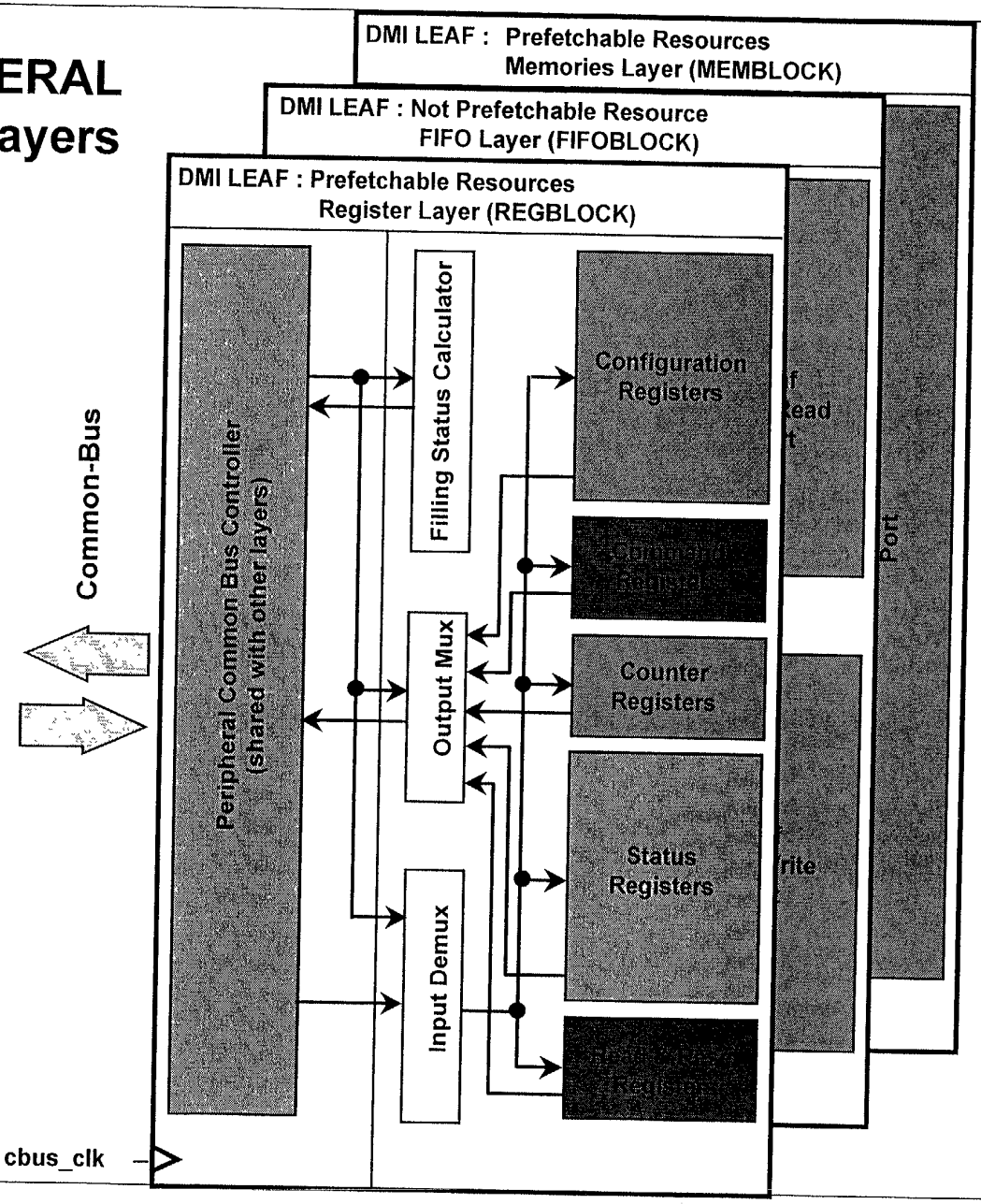


Figure 17

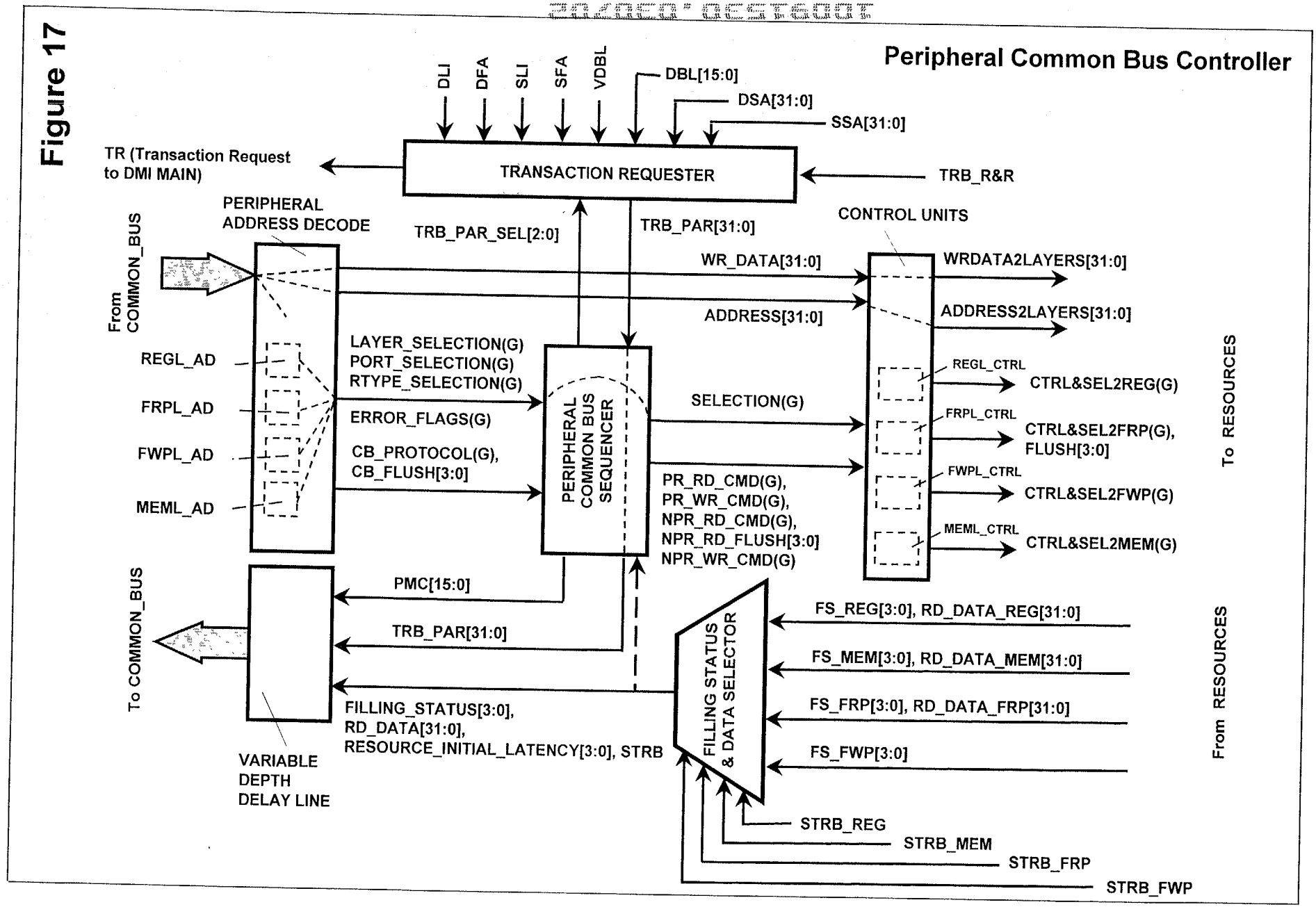

Peripheral Common Bus Controller
DLI
DFA
SLI
SFA
VDBL
DBL[15:0]
DSA[31:0]
SSA[31:0]
TR (Transaction Request to DMI MAIN)
TRANSACTION REQUESTER
TRB_R&R
PERIPHERAL ADDRESS DECODE
TRB_PAR_SEL[2:0]
TRB_PAR[31:0]
CONTROL UNITS
From COMMON_BUS
WR_DATA[31:0]
WRDATA2LAYERS[31:0]
ADDRESS[31:0]
ADDRESS2LAYERS[31:0]
REGL_AD
FRPL_AD
FWPL_AD
MEML_AD
LAYER_SELECTION(G)
PORT_SELECTION(G)
RTYPE_SELECTION(G)
ERROR_FLAGS(G)
CB_PROTOCOL(G),
CB_FLUSH[3:0]
PERIPHERAL COMMON BUS SEQUENCER
SELECTION(G)
PR_RD_CMD(G),
PR_WR_CMD(G),
NPR_RD_CMD(G),
NPR_RD_FLUSH[3:0]
NPR_WR_CMD(G)
REGL_CTRL
CTRL&SEL2REG(G)
FRPL_CTRL
CTRL&SEL2FRP(G),
FLUSH[3:0]
FWPL_CTRL
CTRL&SEL2FWP(G)
MEML_CTRL
CTRL&SEL2MEM(G)
To RESOURCES
To COMMON_BUS
PMC[15:0]
TRB_PAR[31:0]
FILLING_STATUS[3:0],
RD_DATA[31:0],
RESOURCE_INITIAL_LATENCY[3:0], STRB
VARIABLE DEPTH DELAY LINE
FILLING STATUS & DATA SELECTOR
FS_REG[3:0], RD_DATA_REG[31:0]
FS_MEM[3:0], RD_DATA_MEM[31:0]
FS_FRP[3:0], RD_DATA_FRP[31:0]
FS_FWP[3:0]
From RESOURCES
STRB_REG
STRB_MEM
STRB_FRP
STRB_FWP

Figure 18

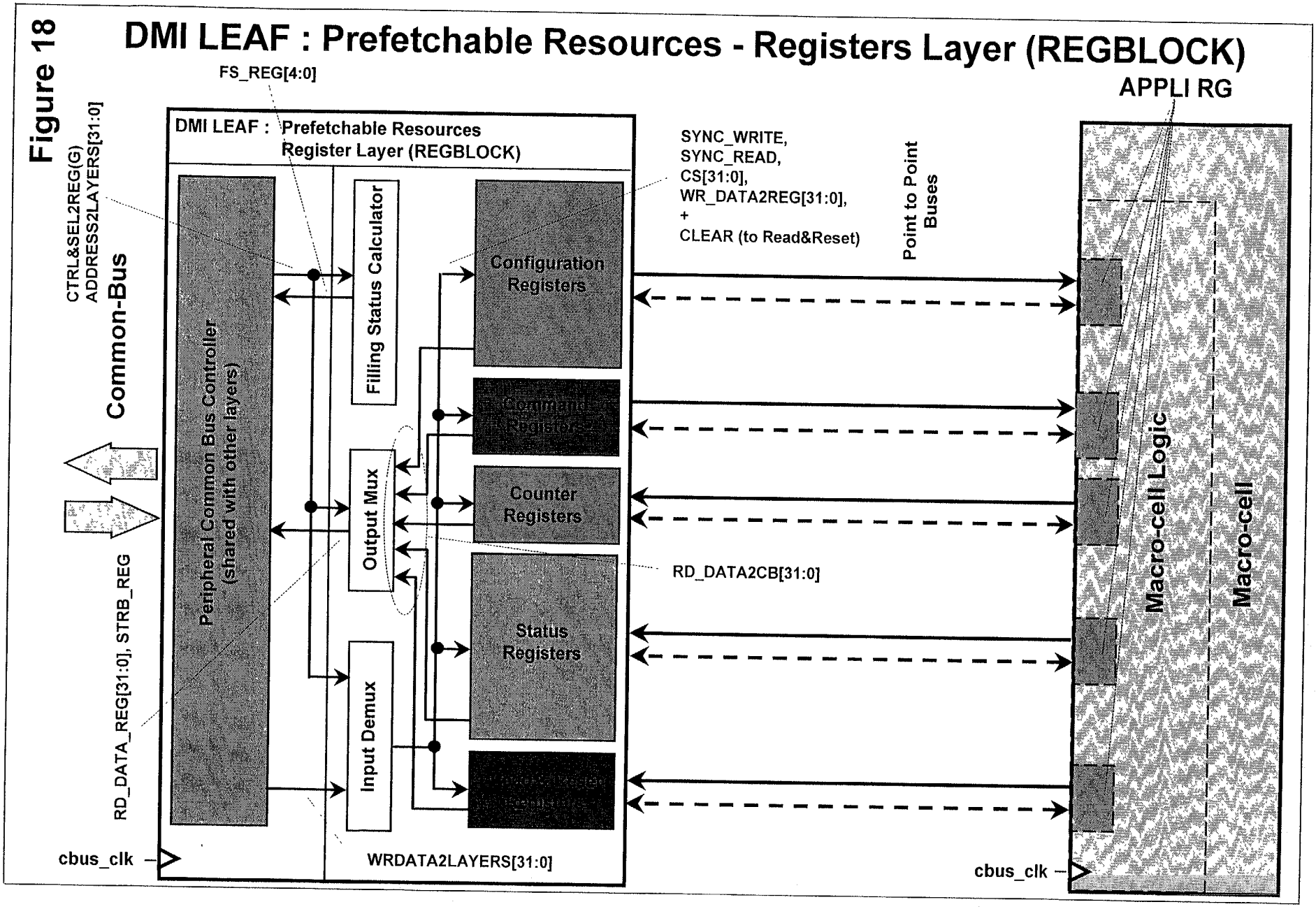

DMI LEAF : Prefetchable Resources - Registers Layer (REGBLOCK)
FS_REG[4:0]
APPLI RG
DMI LEAF : Prefetchable Resources
Register Layer (REGBLOCK)
CTRL&SEL2REG(G)
ADDRESS2LAYERS[31:0]
Common-Bus
SYNC_WRITE,
SYNC_READ,
CS[31:0],
WR_DATA2REG[31:0],
+
CLEAR (to Read&Reset)
Point to Point
Buses
Peripheral Common Bus Controller
(shared with other layers)
Filling Status Calculator
Configuration
Registers
Command
Registers
Output Mux
Counter
Registers
Status
Registers
Input Demux
RD_DATA2CB[31:0]
RD_DATA_REG[31:0], STRB_REG
cbus_clk
WRDATA2LAYERS[31:0]
Macro-cell Logic
Macro-cell
cbus_clk

Figure 19

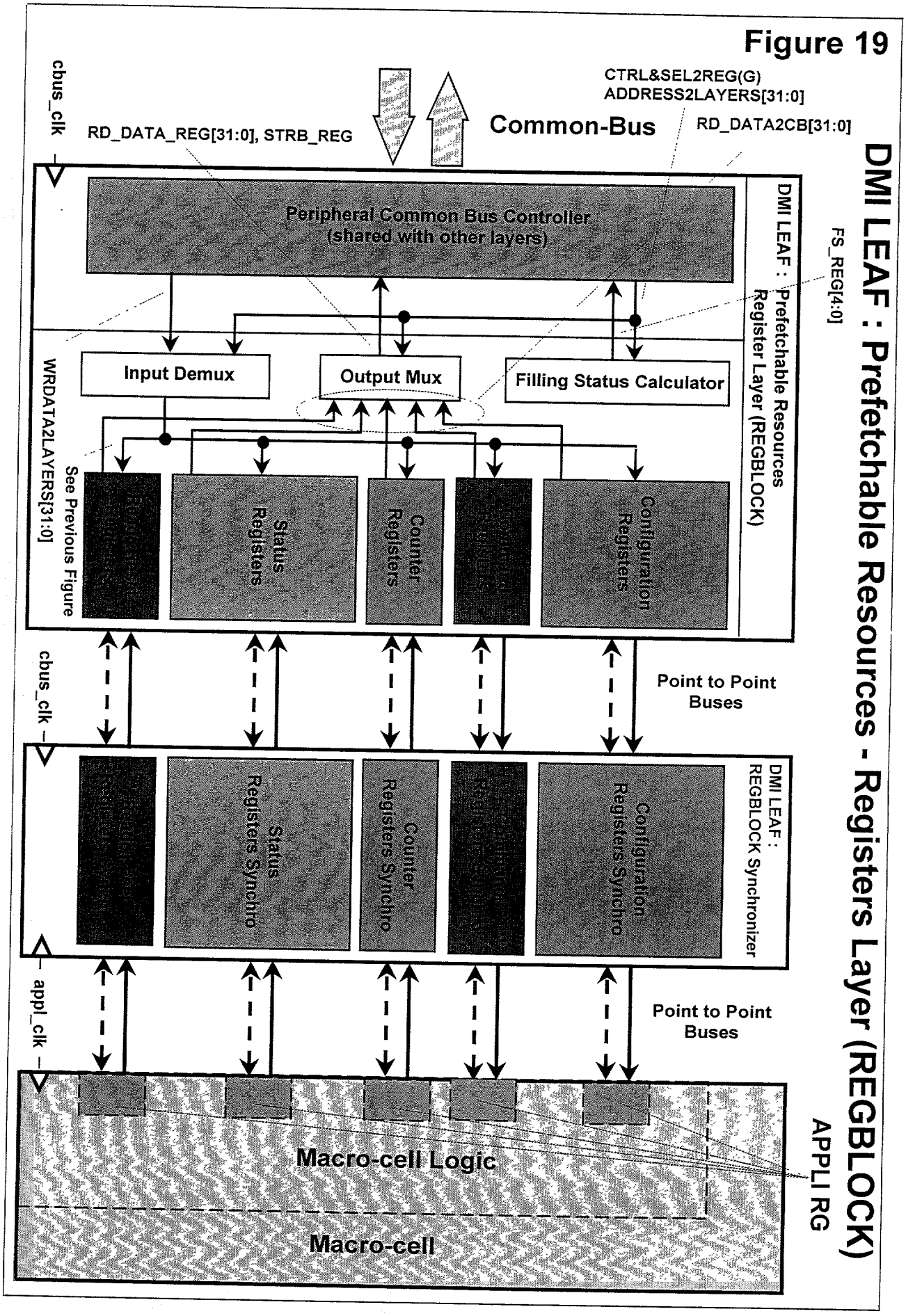
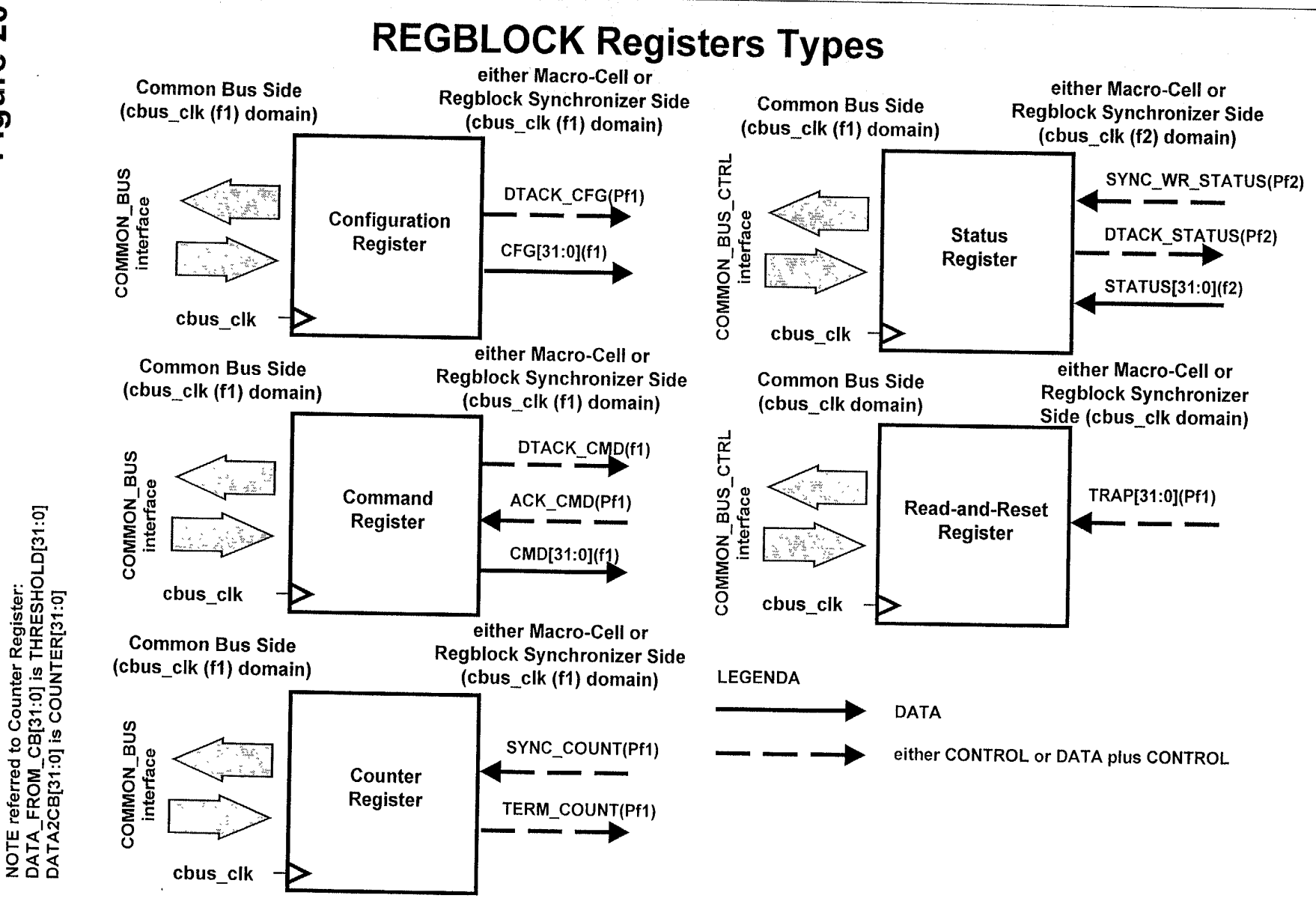


Figure 20



NOTE referred to Counter Register:
DATA_FROM_CB[31:0] is THRESHOLD[31:0]
DATA2CB[31:0] is COUNTER[31:0]

Figure 21

REGBLOCK SYNCHRONIZER Register Synchronizer Types

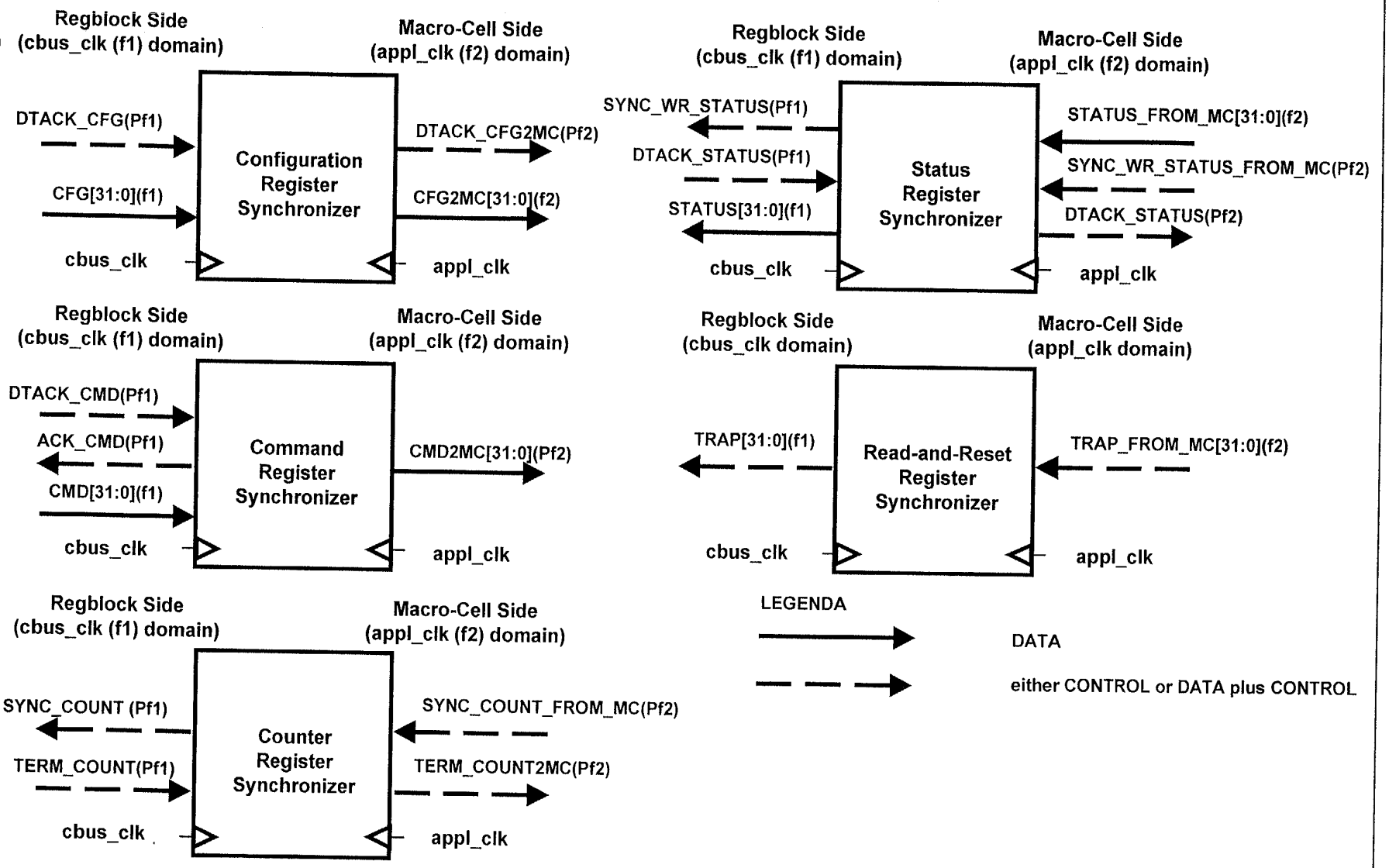


Figure 22

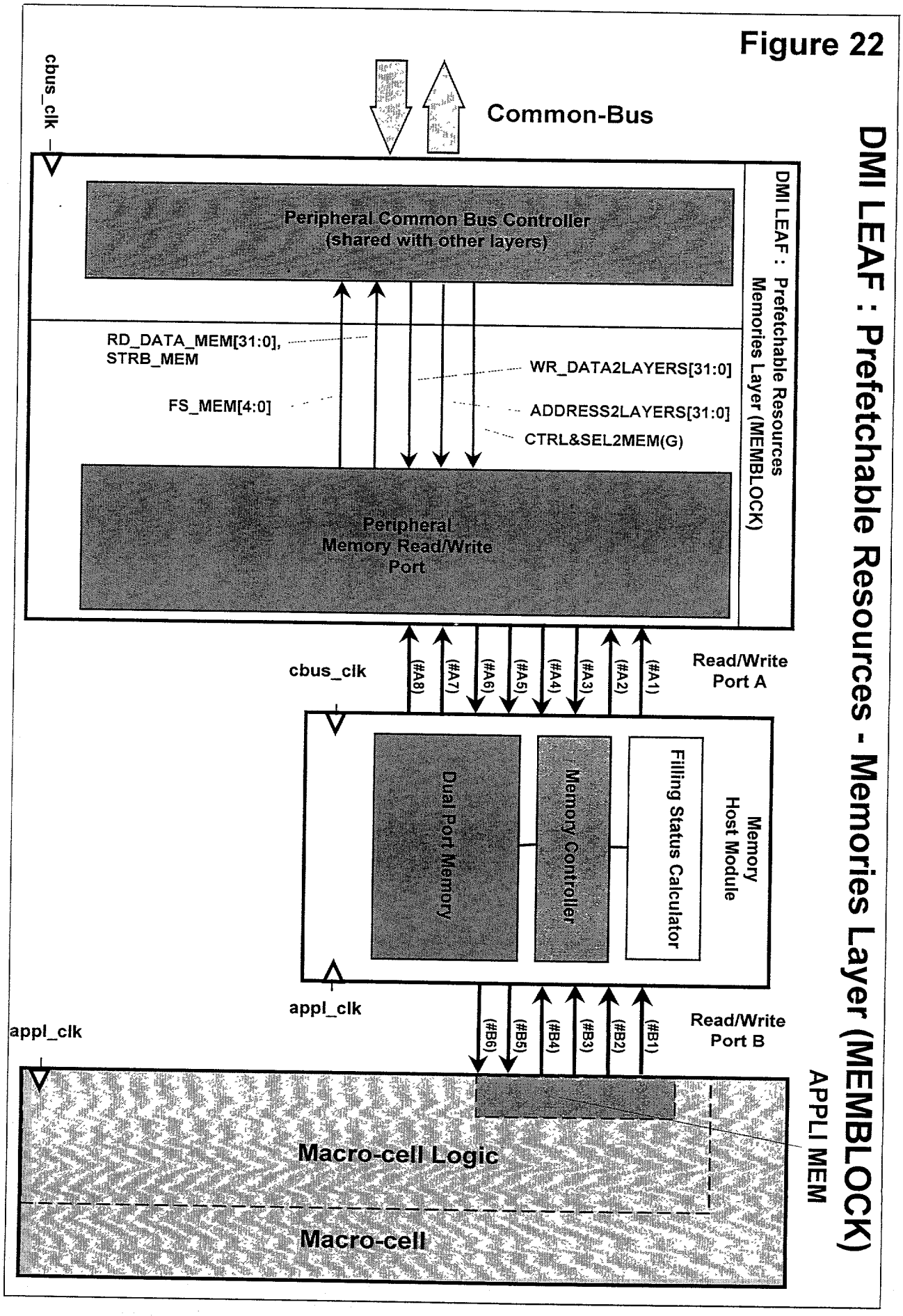
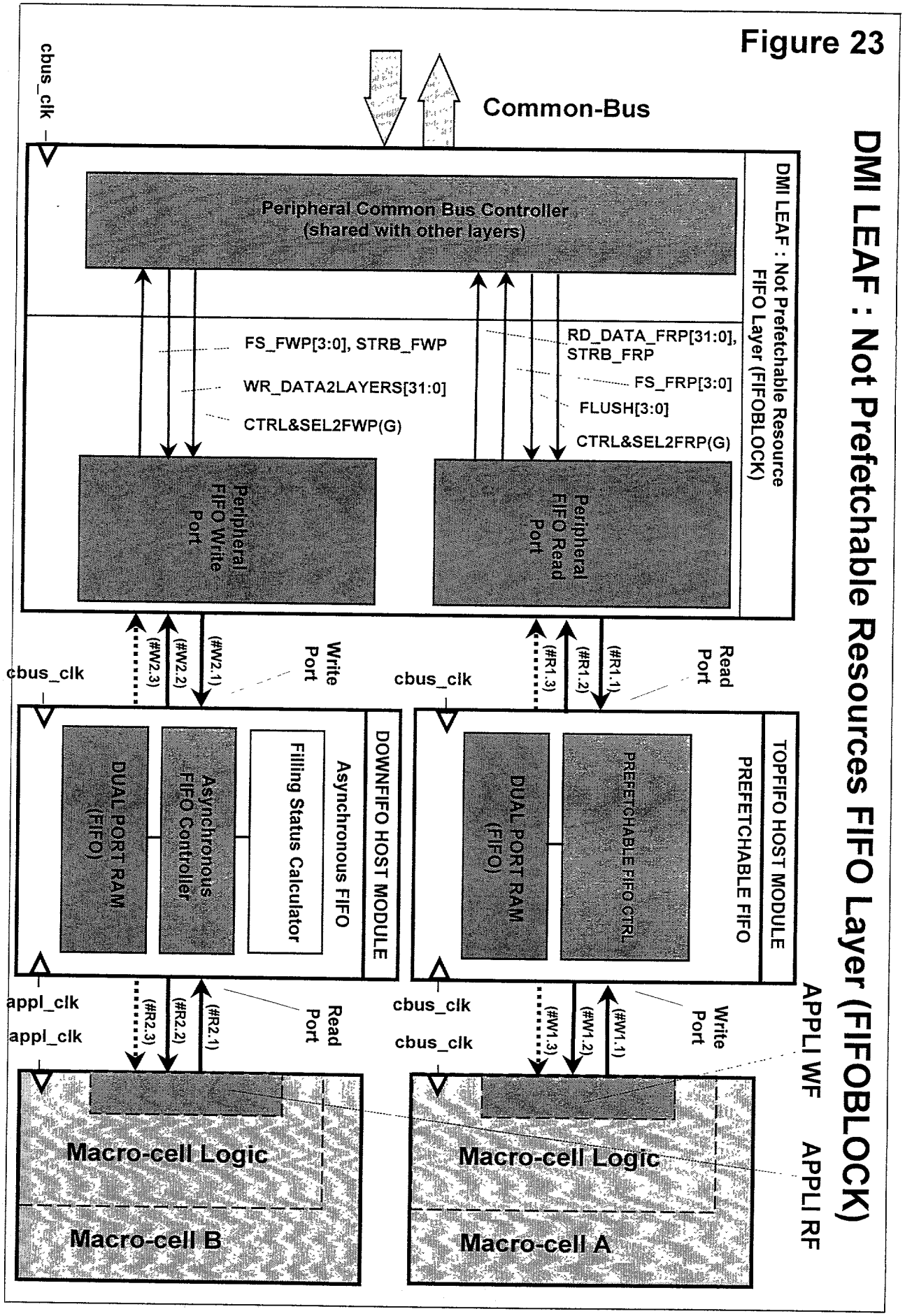

DMI LEAF : Prefetchable Resources - Memories Layer (MEMBLOCK)
Common-Bus
cbus_clk
Peripheral Common Bus Controller (shared with other layers)
DMI LEAF : Prefetchable Resources Memories Layer (MEMBLOCK)
RD_DATA_MEM[31:0], STRB_MEM
WR_DATA2LAYERS[31:0]
FS_MEM[4:0]
ADDRESS2LAYERS[31:0]
CTRL&SEL2MEM(G)
Peripheral Memory Read/Write Port
cbus_clk
(#A8)
(#A7)
(#A6)
(#A5)
(#A4)
(#A3)
(#A2)
(#A1)
Read/Write Port A
Memory Host Module
Dual Port Memory
Memory Controller
Filling Status Calculator
appl_clk
appl_clk
(#B6)
(#B5)
(#B4)
(#B3)
(#B2)
(#B1)
Read/Write Port B
APPLI MEM
Macro-cell Logic
Macro-cell

Figure 23



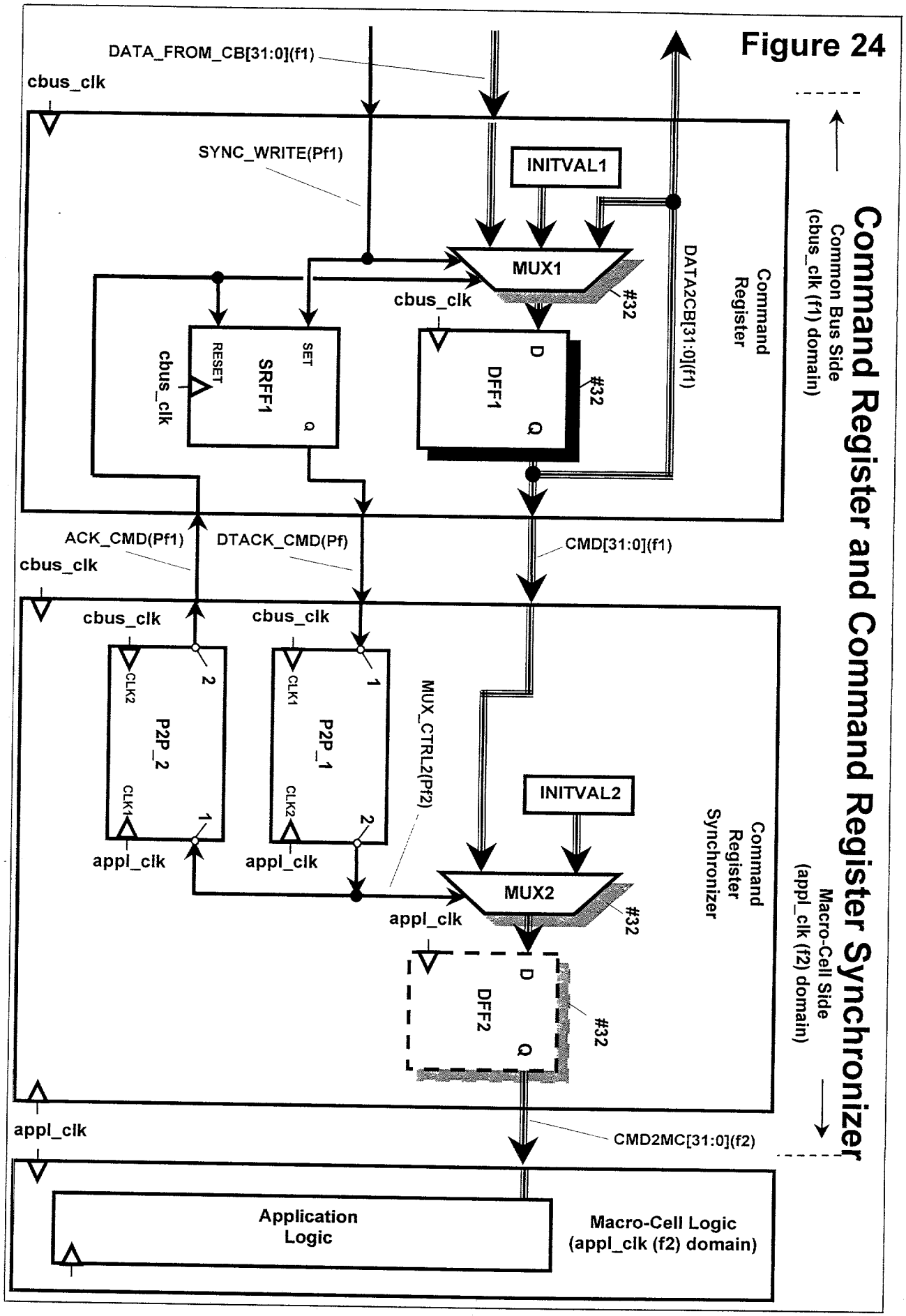


Figure 24
Command Register and Command Register Synchronizer
Common Bus Side
(cbus_clk (f1) domain)
Macro-Cell Side
(appl_clk (f2) domain)
DATA_FROM_CB[31:0](f1)
cbus_clk
SYNC_WRITE(Pf1)
INITVAL1
MUX1
#32
cbus_clk
DATA2CB[31:0](f1)
Command Register
SRFF1
RESET
SET
Q
DFF1
D
Q
#32
ACK_CMD(Pf1)
DTACK_CMD(Pf)
CMD[31:0](f1)
cbus_clk
P2P_2
CLK2
CLK1
2
1
P2P_1
CLK1
CLK2
1
2
MUX_CTRL2(Pf2)
INITVAL2
MUX2
#32
appl_clk
DFF2
D
Q
#32
Command Register Synchronizer
appl_clk
CMD2MC[31:0](f2)
Application Logic
Macro-Cell Logic
(appl_clk (f2) domain)

Figure 25

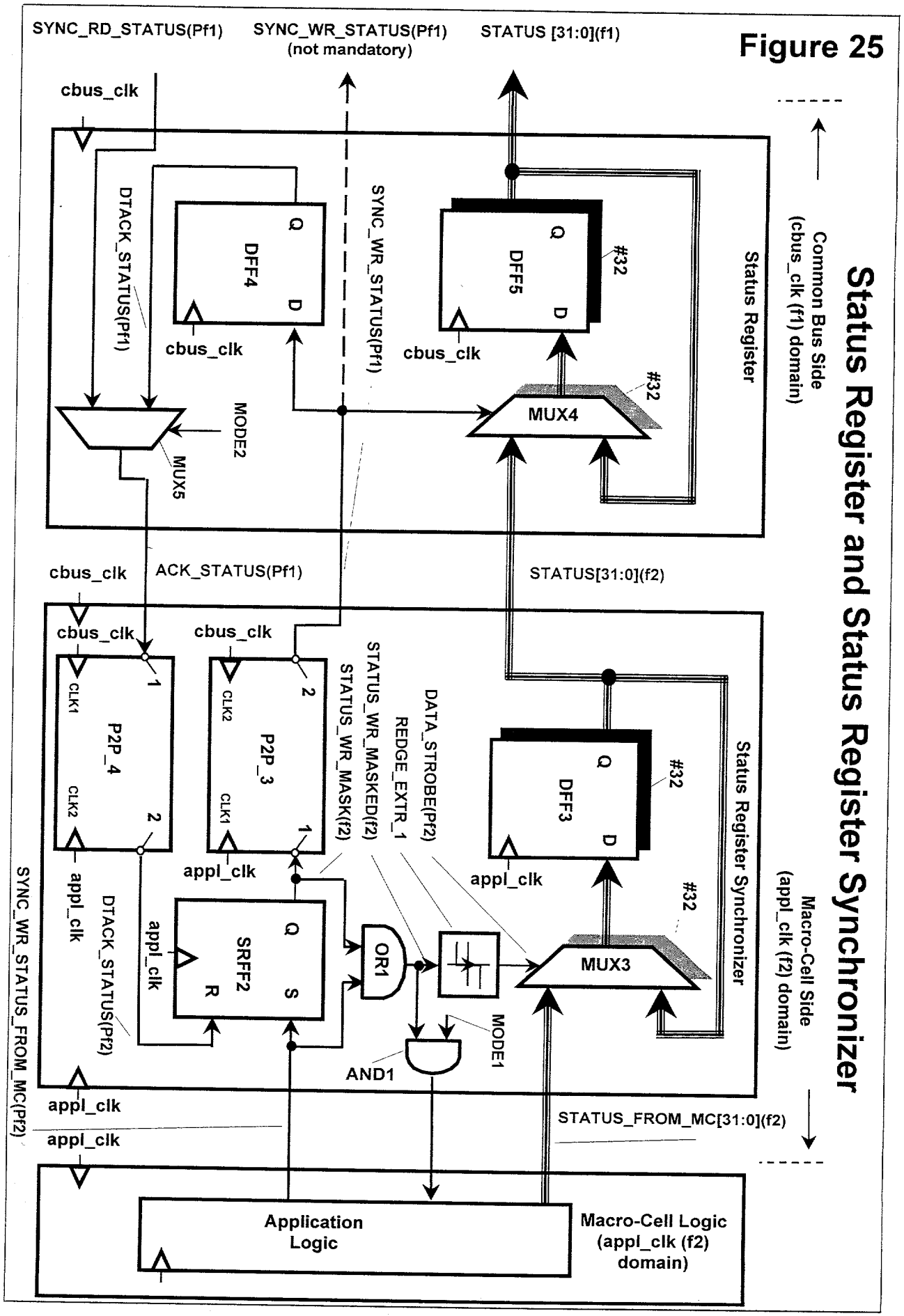

Status Register and Status Register Synchronizer
SYNC_RD_STATUS(Pf1)
SYNC_WR_STATUS(Pf1)
(not mandatory)
STATUS [31:0](f1)
Common Bus Side
(cbus_clk (f1) domain)
cbus_clk
DTACK_STATUS(Pf1)
SYNC_WR_STATUS(Pf1)
DFF4
DFF5
#32
Status Register
MUX4
MODE2
MUX5
ACK_STATUS(Pf1)
STATUS[31:0](f2)
P2P_4
P2P_3
CLK1
CLK2
STATUS_WR_MASK(f2)
STATUS_WR_MASKED(f2)
DATA_STROBE(Pf2)
REDGE_EXTR_1
DFF3
Status Register Synchronizer
Macro-Cell Side
(appl_clk (f2) domain)
appl_clk
SRFF2
OR1
MUX3
AND1
MODE1
DTACK_STATUS(Pf2)
SYNC_WR_STATUS_FROM_MC(Pf2)
STATUS_FROM_MC[31:0](f2)
Application
Logic
Macro-Cell Logic
(appl_clk (f2)
domain)

Figure 26

Pulse to Pulse Synchronization Unit

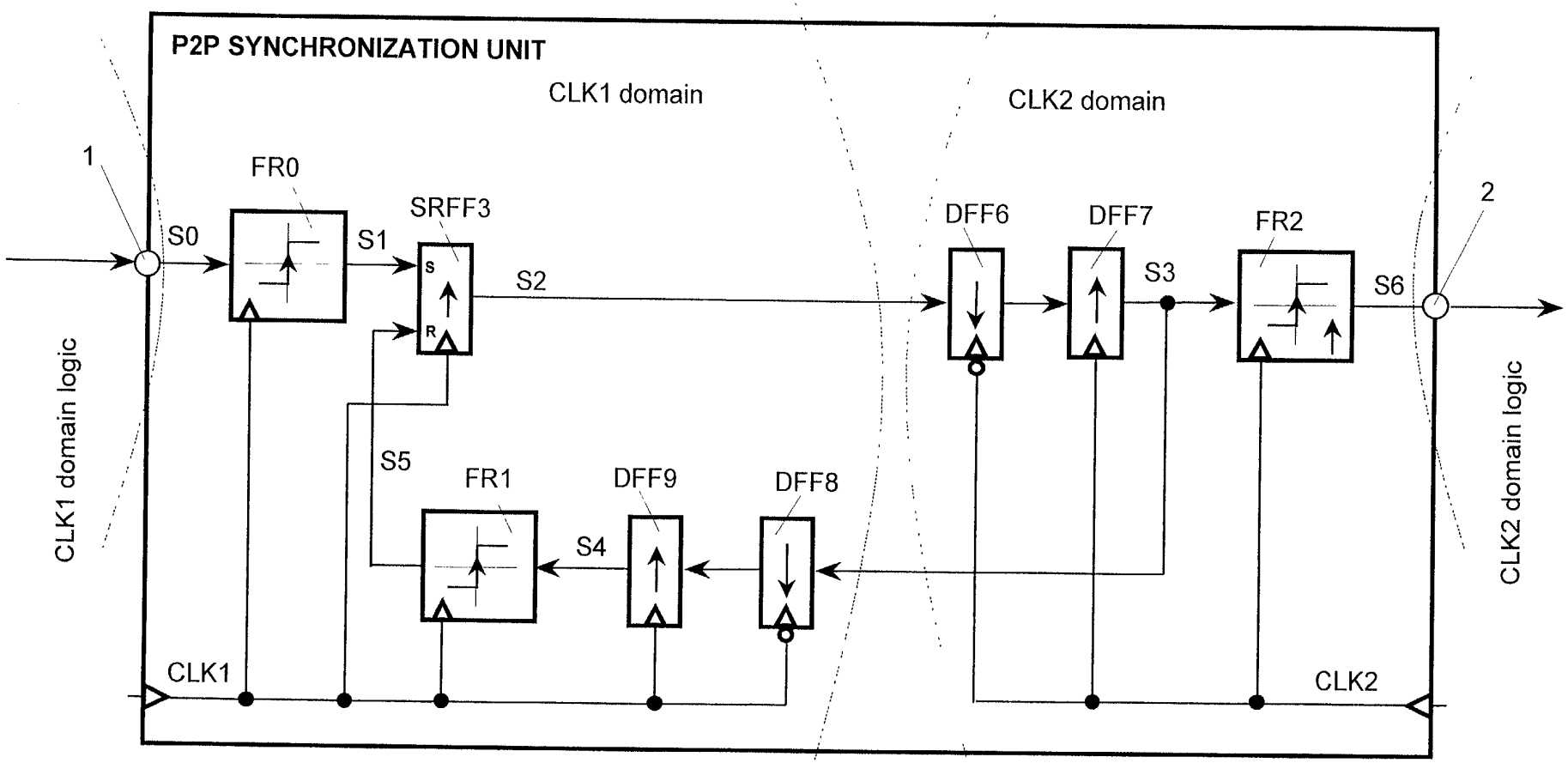


Figure 27

Timing Diagram of Pulse to Pulse Synchronization Unit

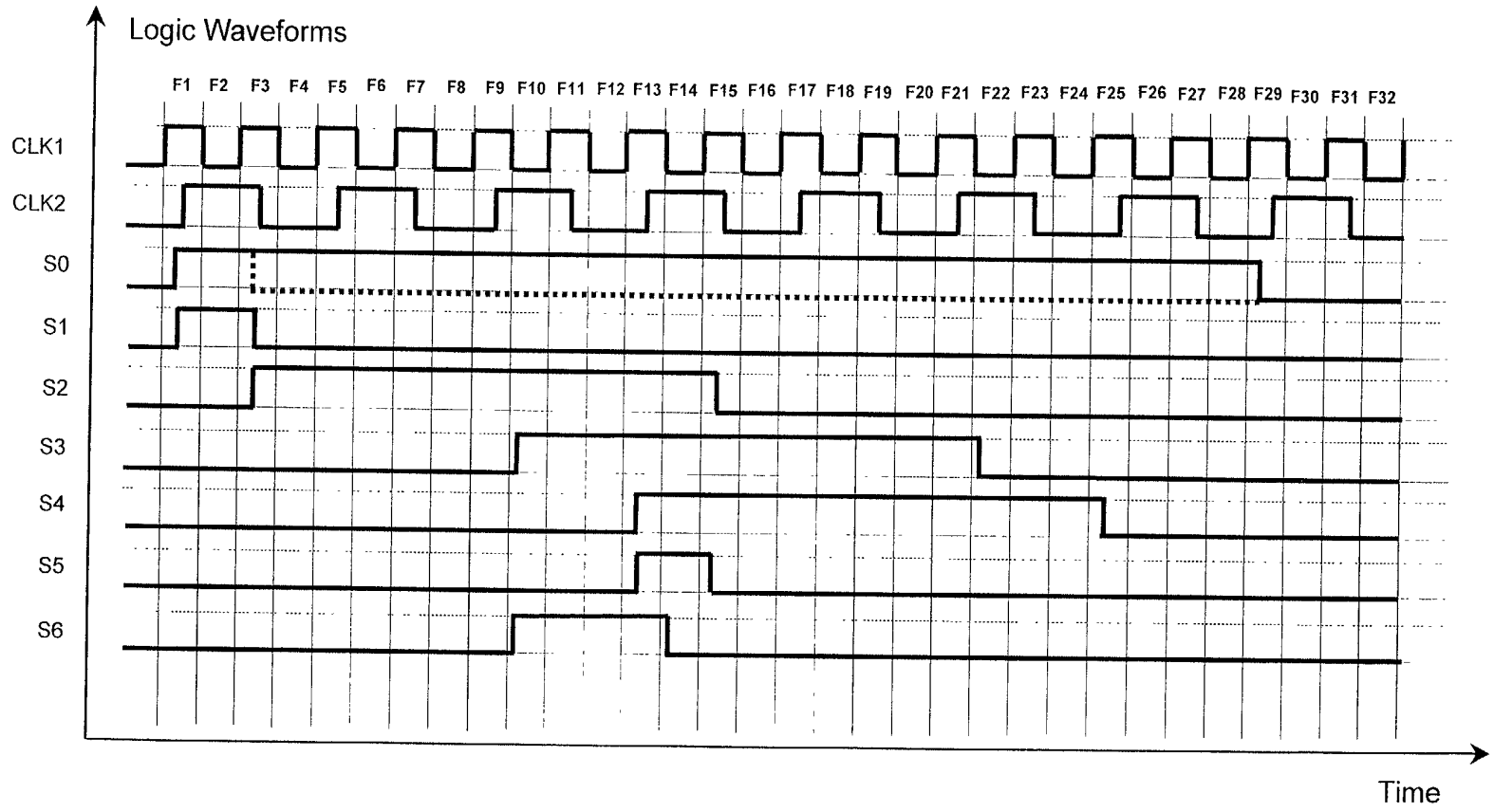


Figure 28

Advantages of Distributed Synchronization

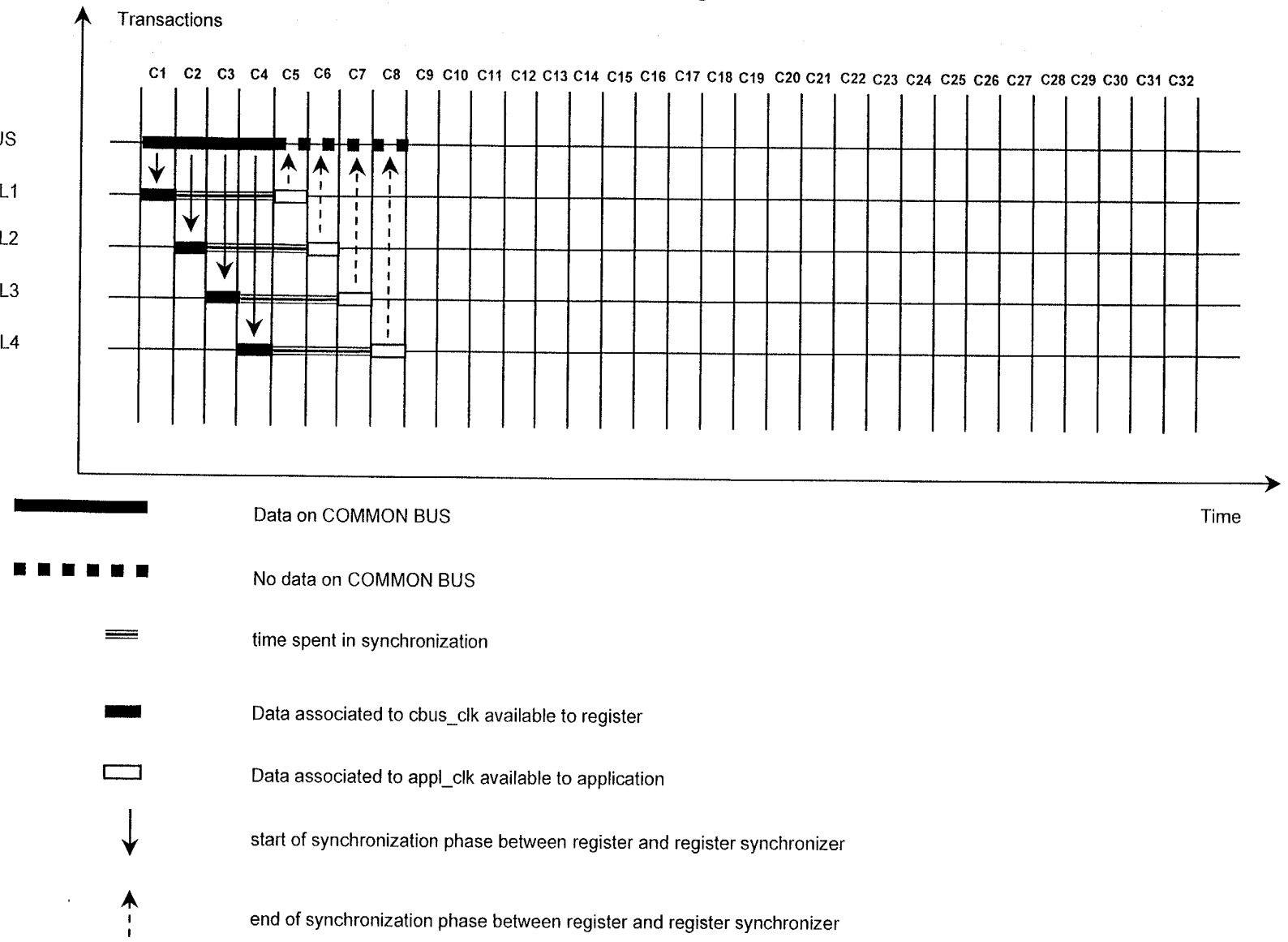


Figure29

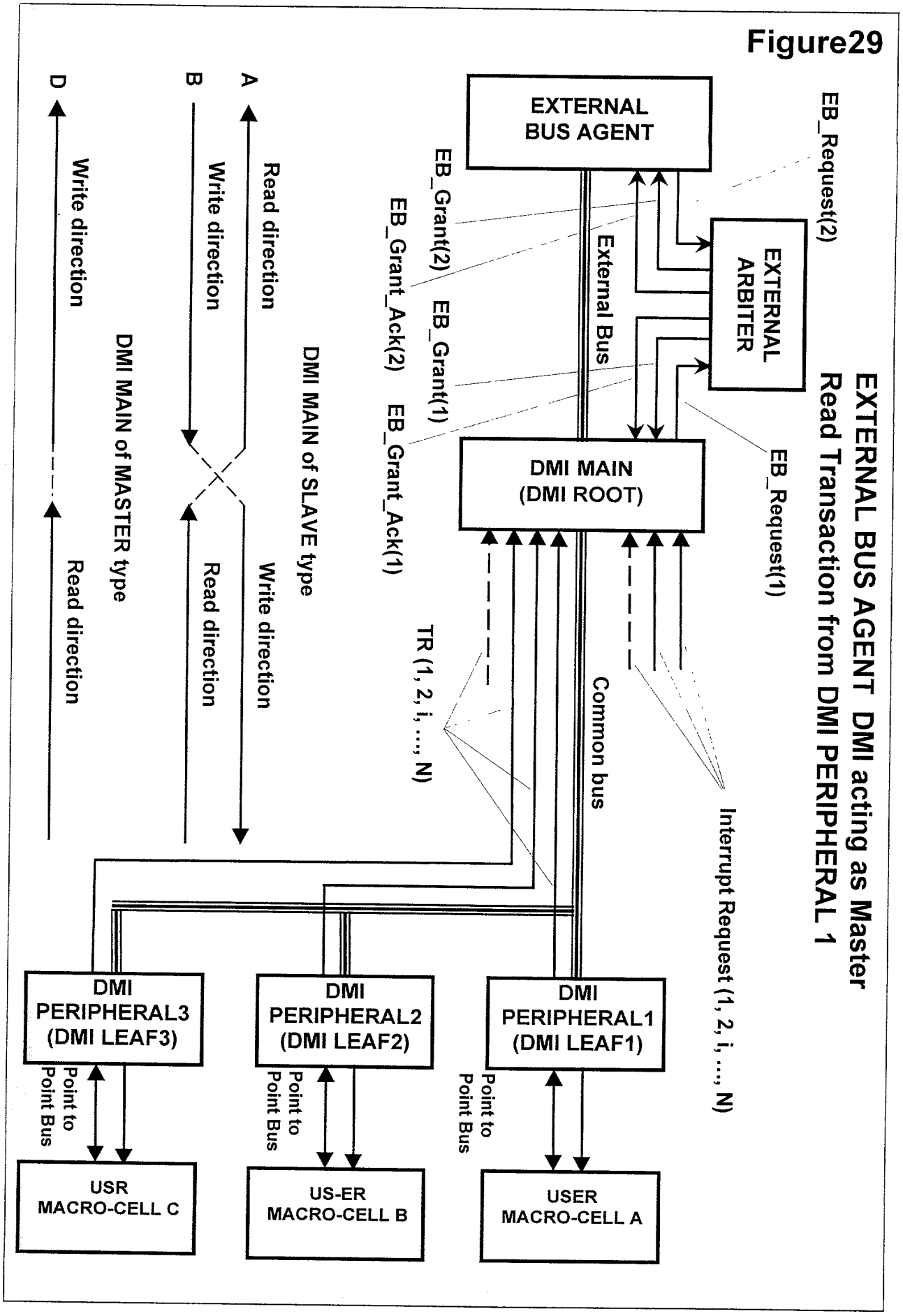

EXTERNAL BUS AGENT DMI acting as Master
Read Transaction from DMI PERIPHERAL 1
EXTERNAL BUS AGENT
EXTERNAL ARBITER
DMI MAIN (DMI ROOT)
EB_Request(2)
EB_Grant(2)
EB_Grant_Ack(2)
EB_Grant(1)
EB_Grant_Ack(1)
EB_Request(1)
External Bus
Common bus
TR (1, 2, i, ..., N)
Interrupt Request (1, 2, i, ..., N)
A
Read direction
B
Write direction
DMI MAIN of SLAVE type
DMI MAIN of MASTER type
D
Write direction
C
Read direction
DMI PERIPHERAL3 (DMI LEAF3)
DMI PERIPHERAL2 (DMI LEAF2)
DMI PERIPHERAL1 (DMI LEAF1)
Point to Point Bus
Point to Point Bus
Point to Point Bus
USR MACRO-CELL C
US-ER MACRO-CELL B
USER MACRO-CELL A

Figure 30

Prior Art

Asynchronous two phase handshake protocol: read

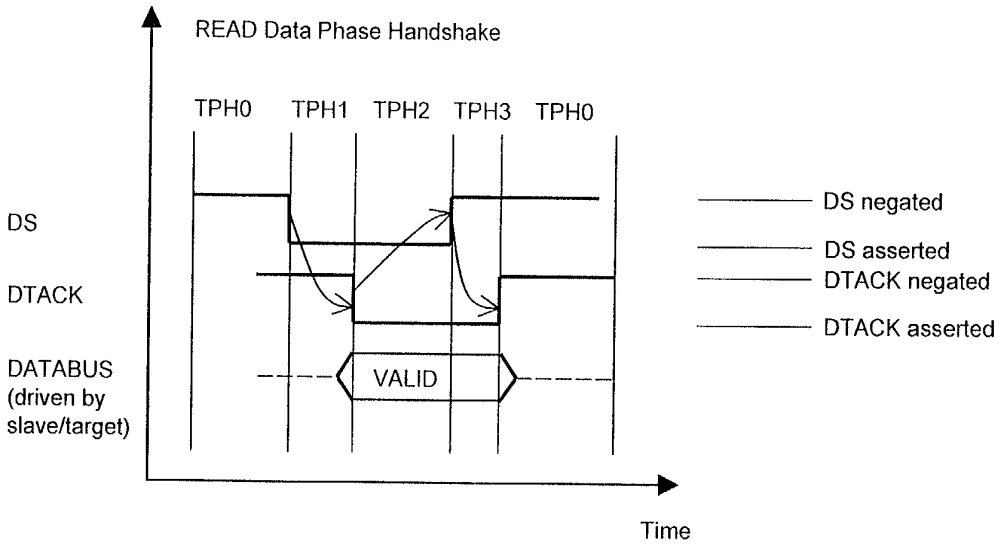


Figure 31

Prior Art

Asynchronous two phase handshake protocol: write

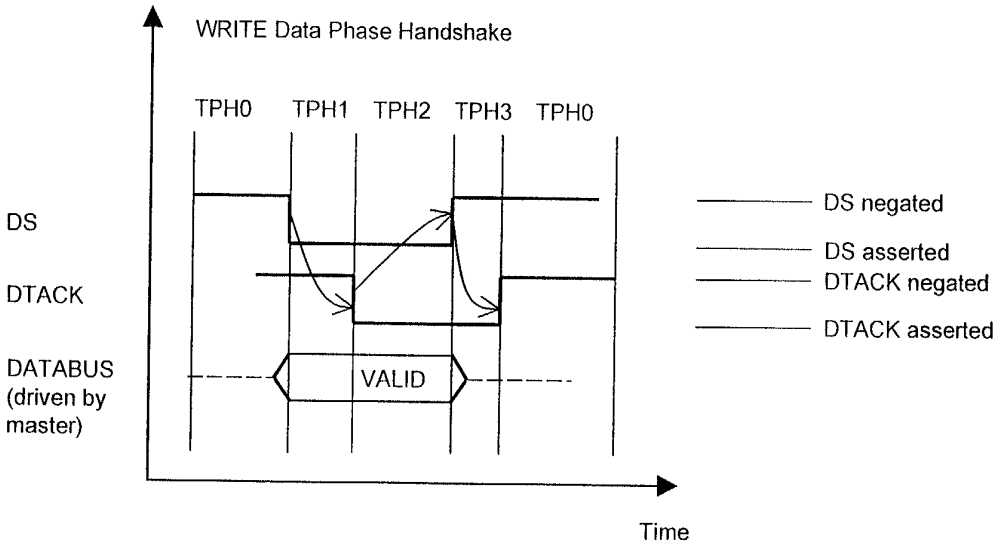
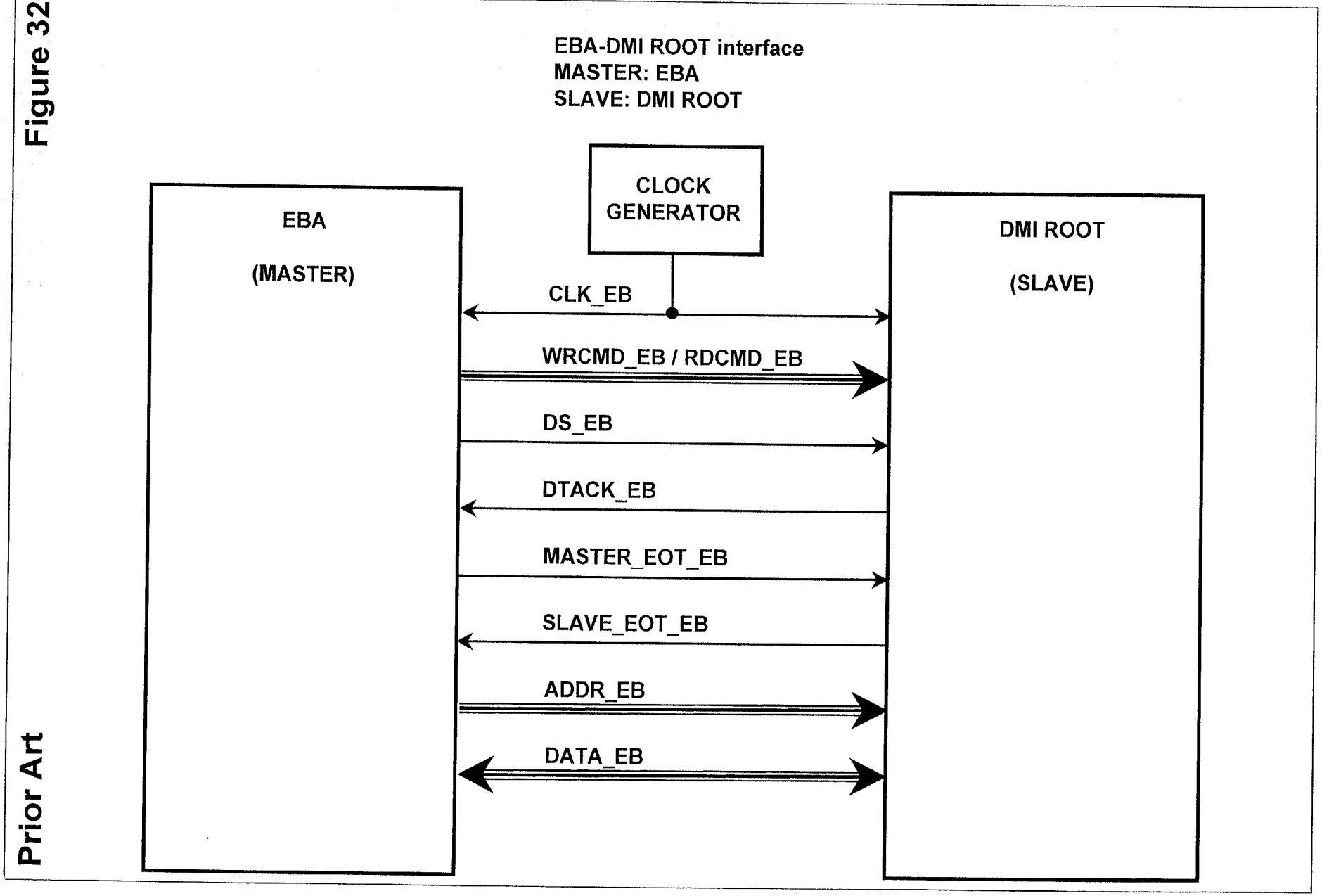


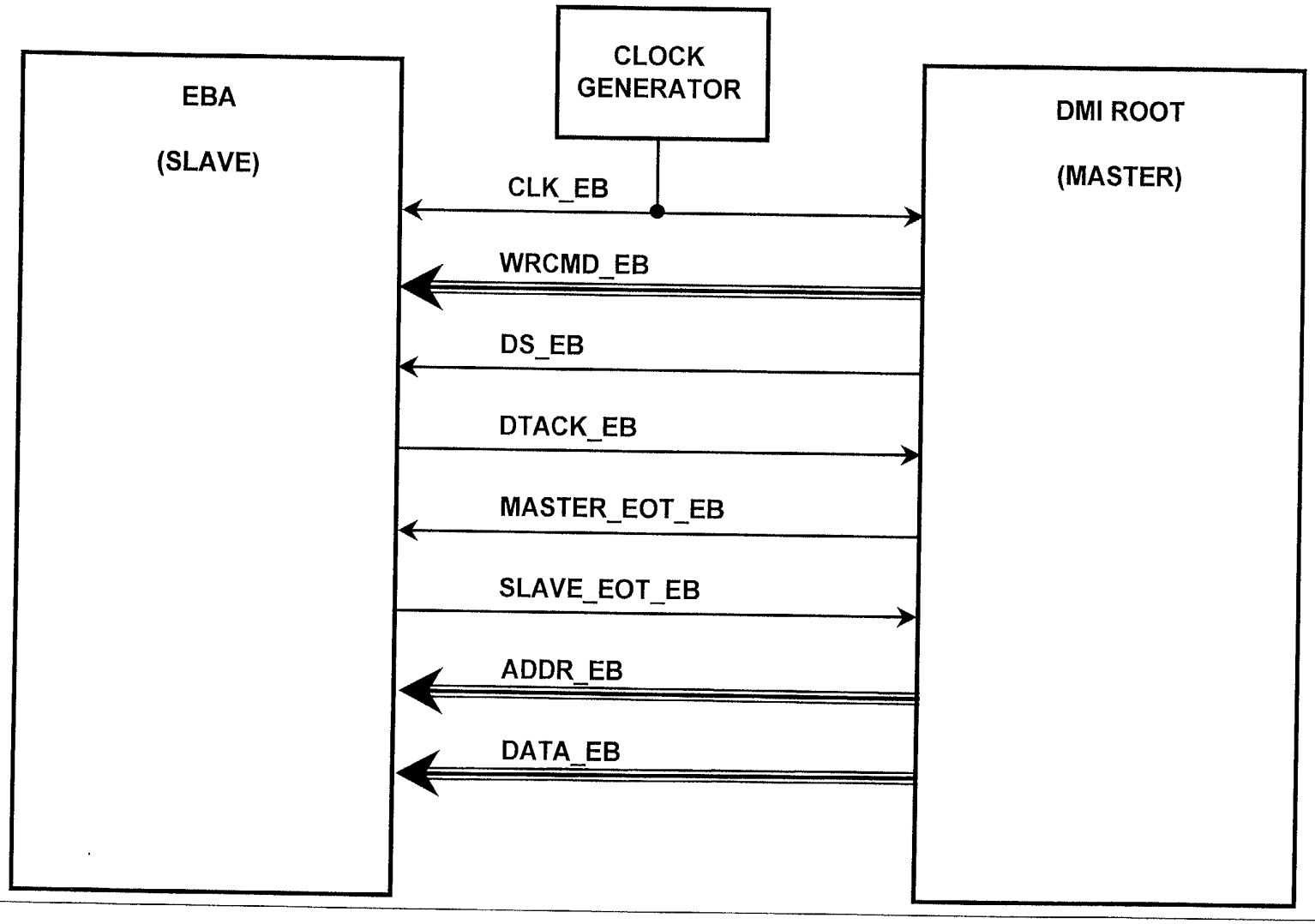
Figure 32



Prior Art

Figure 33

EBA-DMI ROOT interface
MASTER: DMI ROOT
SLAVE: EBA



Prior Art

Figure 34

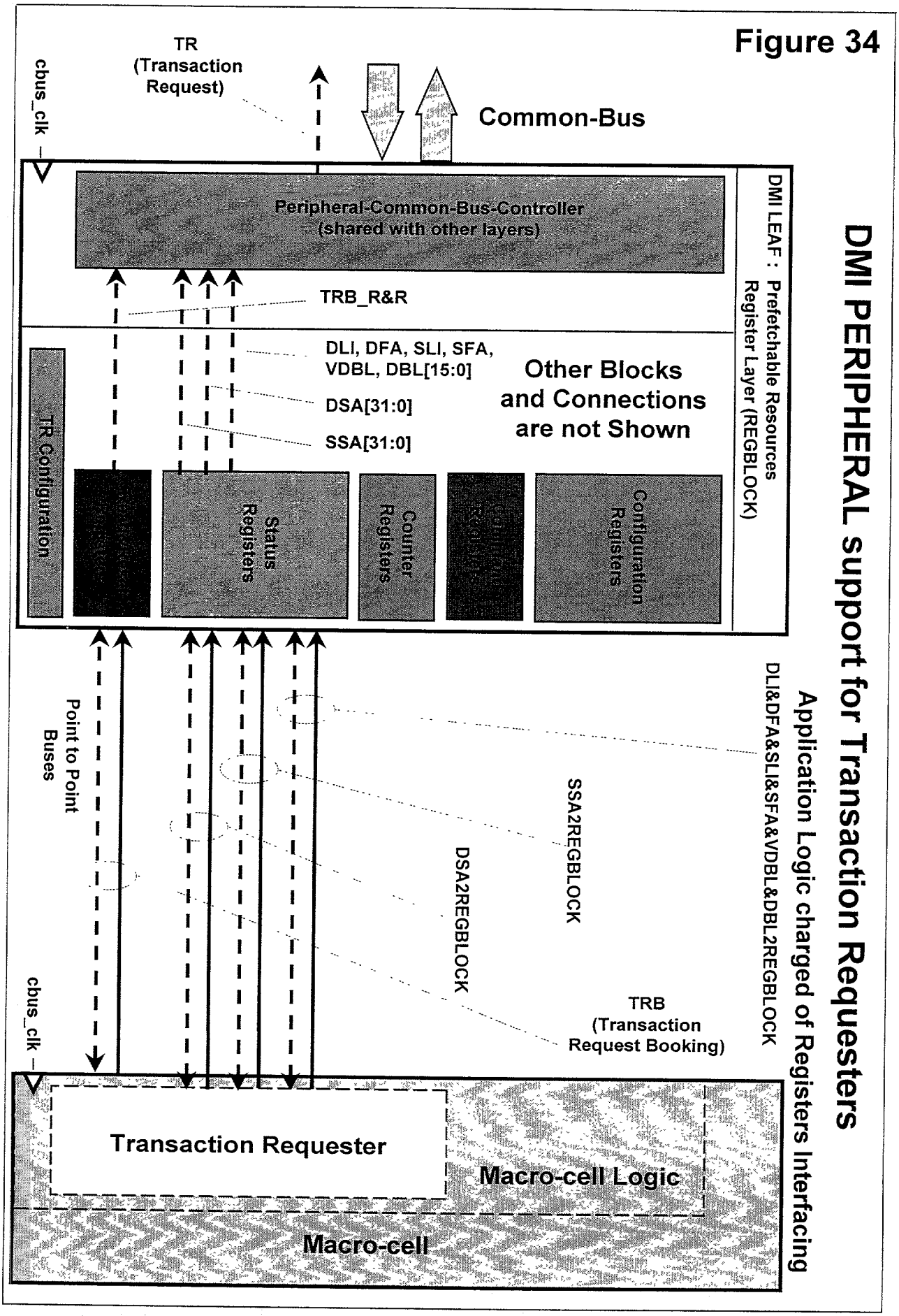

DMI PERIPHERAL support for Transaction Requesters
TR
(Transaction
Request)
Common-Bus
cbus_clk
Peripheral-Common-Bus-Controller
(shared with other layers)
DMI LEAF : Prefetchable Resources
Register Layer (REGBLOCK)
TRB_R&R
DLI, DFA, SLI, SFA,
VDBL, DBL[15:0]
DSA[31:0]
SSA[31:0]
Other Blocks
and Connections
are not Shown
TR Configuration
Status
Registers
Counter
Registers
Configuration
Registers
Point to Point
Buses
Application Logic charged of Registers Interfacing
DLI&DFA&SLI&SFA&VDBL&DBL2REGBLOCK
SSA2REGBLOCK
DSA2REGBLOCK
TRB
(Transaction
Request Booking)
cbus_clk
Transaction Requester
Macro-cell Logic
Macro-cell

Figure 35

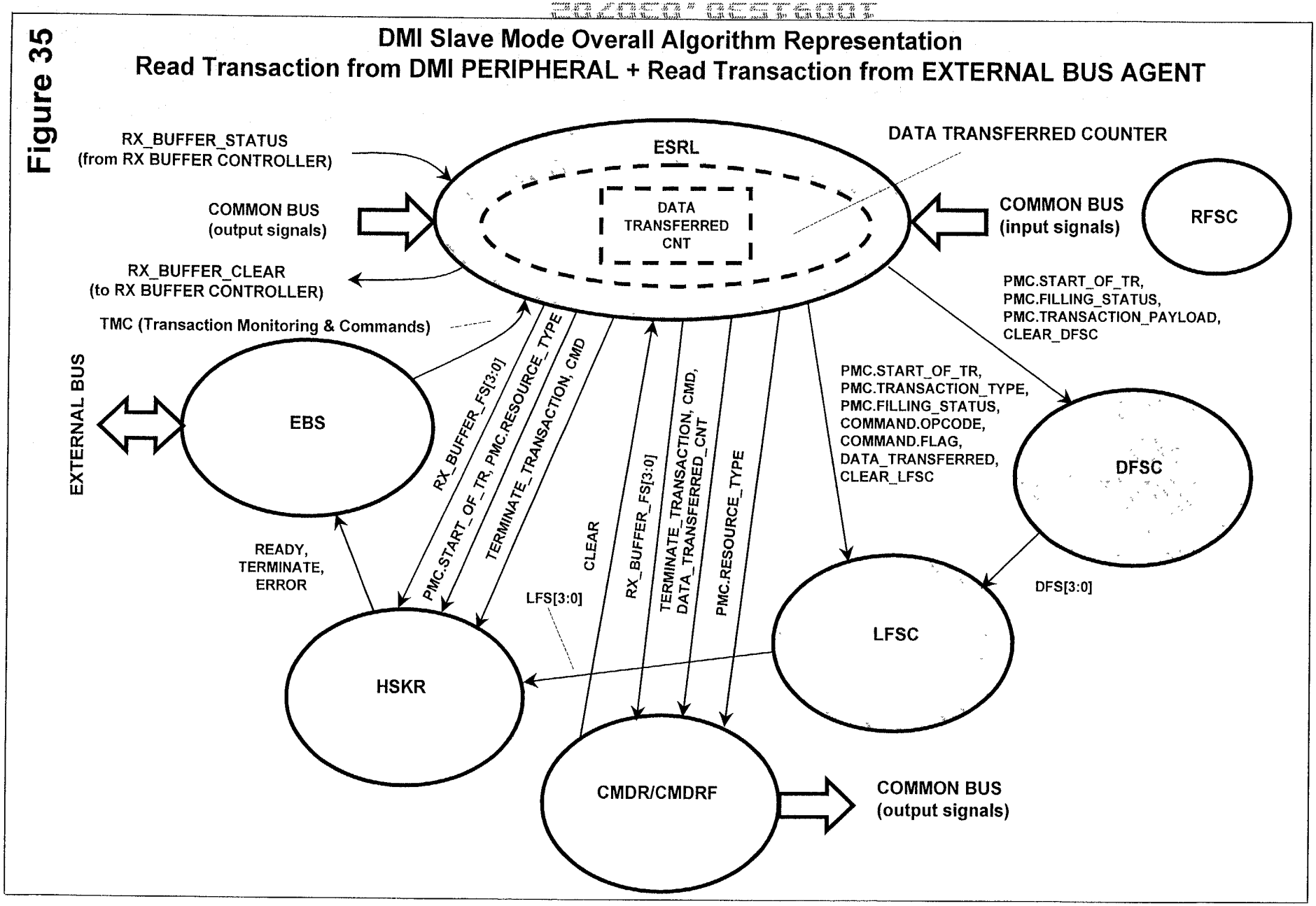

DMI Slave Mode Overall Algorithm Representation
Read Transaction from DMI PERIPHERAL + Read Transaction from EXTERNAL BUS AGENT
RX_BUFFER_STATUS
(from RX BUFFER CONTROLLER)
COMMON BUS
(output signals)
RX_BUFFER_CLEAR
(to RX BUFFER CONTROLLER)
TMC (Transaction Monitoring & Commands)
EXTERNAL BUS
ESRL
DATA
TRANSFERRED
CNT
DATA TRANSFERRED COUNTER
COMMON BUS
(input signals)
RFSC
PMC.START_OF_TR,
PMC.FILLING_STATUS,
PMC.TRANSACTION_PAYLOAD,
CLEAR_DFSC
EBS
RX_BUFFER_FS[3:0]
PMC.START_OF_TR, PMC.RESOURCE_TYPE
TERMINATE_TRANSACTION, CMD
CLEAR
RX_BUFFER_FS[3:0]
TERMINATE_TRANSACTION, CMD,
DATA_TRANSFERRED_CNT
PMC.RESOURCE_TYPE
PMC.START_OF_TR,
PMC.TRANSACTION_TYPE,
PMC.FILLING_STATUS,
COMMAND.OPCODE,
COMMAND.FLAG,
DATA_TRANSFERRED,
CLEAR_LFSC
DFSC
READY,
TERMINATE,
ERROR
LFS[3:0]
DFS[3:0]
LFSC
HSKR
CMDR/CMDRF
COMMON BUS
(output signals)

Figure 36

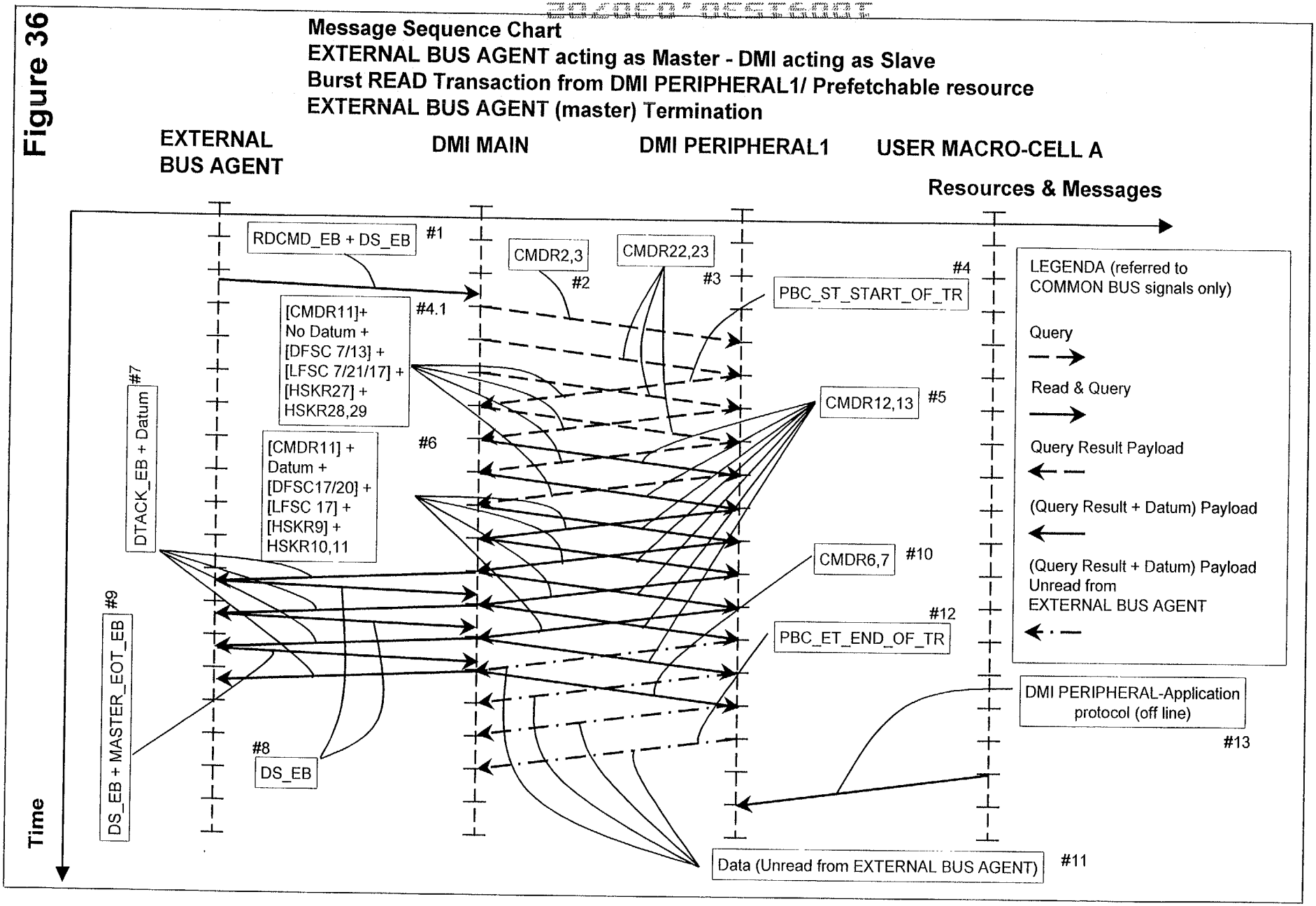

Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
Burst READ Transaction from DMI PERIPHERAL1/ Prefetchable resource
EXTERNAL BUS AGENT (master) Termination
EXTERNAL BUS AGENT
DMI MAIN
DMI PERIPHERAL1
USER MACRO-CELL A
Resources & Messages
Time
RDCMD_EB + DS_EB
#1
CMDR2,3
#2
CMDR22,23
#3
PBC_ST_START_OF_TR
#4
[CMDR11]+ No Datum + [DFSC 7/13] + [LFSC 7/21/17] + [HSKR27] + HSKR28,29
#4.1
CMDR12,13
#5
[CMDR11] + Datum + [DFSC17/20] + [LFSC 17] + [HSKR9] + HSKR10,11
#6
DTACK_EB + Datum
#7
CMDR6,7
#10
PBC_ET_END_OF_TR
#12
DS_EB + MASTER_EOT_EB
#9
DS_EB
#8
DMI PERIPHERAL-Application protocol (off line)
#13
Data (Unread from EXTERNAL BUS AGENT)
#11
LEGENDA (referred to COMMON BUS signals only)
Query
Read & Query
Query Result Payload
(Query Result + Datum) Payload
(Query Result + Datum) Payload Unread from EXTERNAL BUS AGENT

Figure 37

Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
Burst READ Transaction from DMI PERIPHERAL1/ Prefetchable resource
DMI (slave) Termination

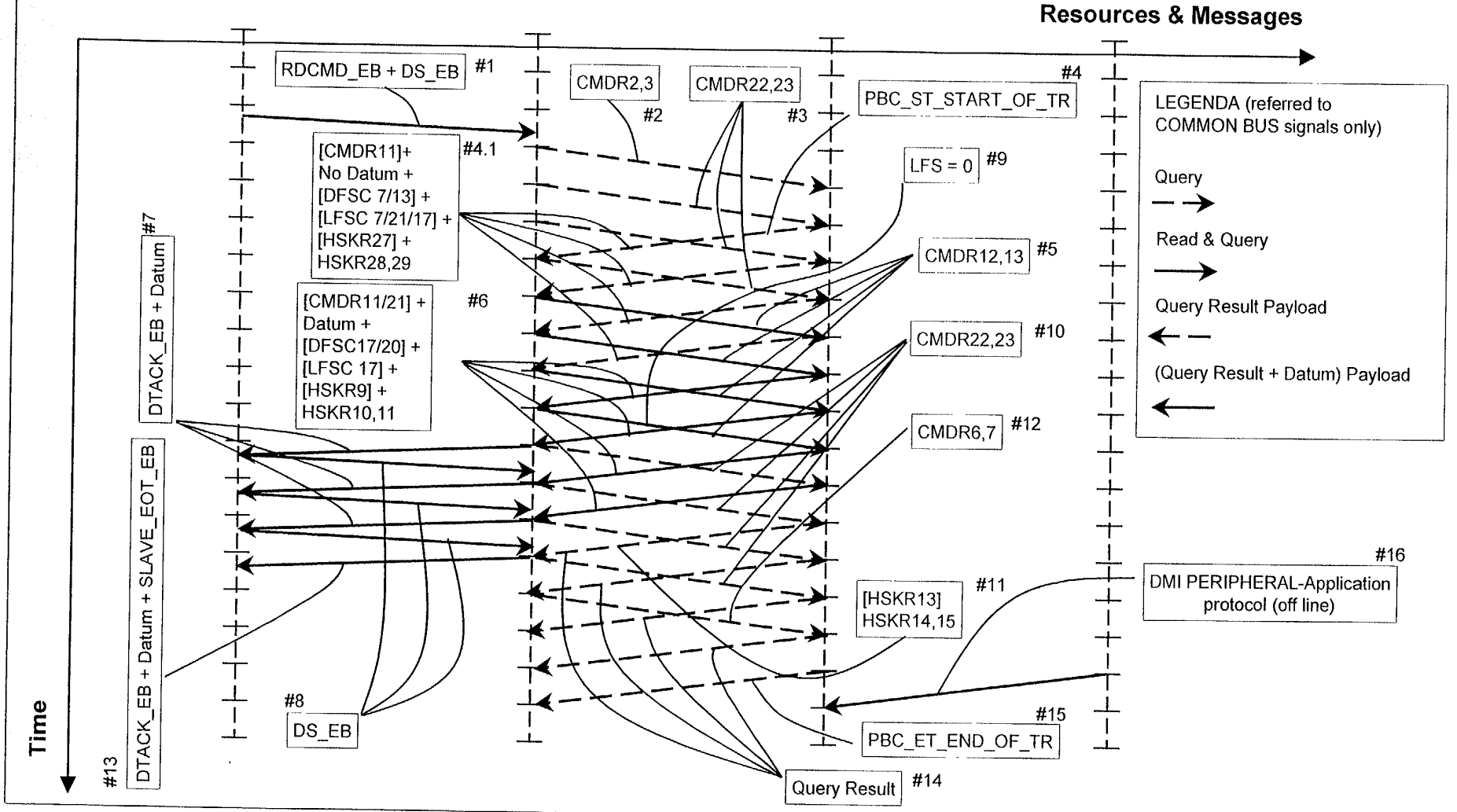


Figure 38

DMI Slave Mode Overall Algorithm Representation
Write Transaction from EXTERNAL BUS AGENT + Write Transaction to DMI PERIPHERAL

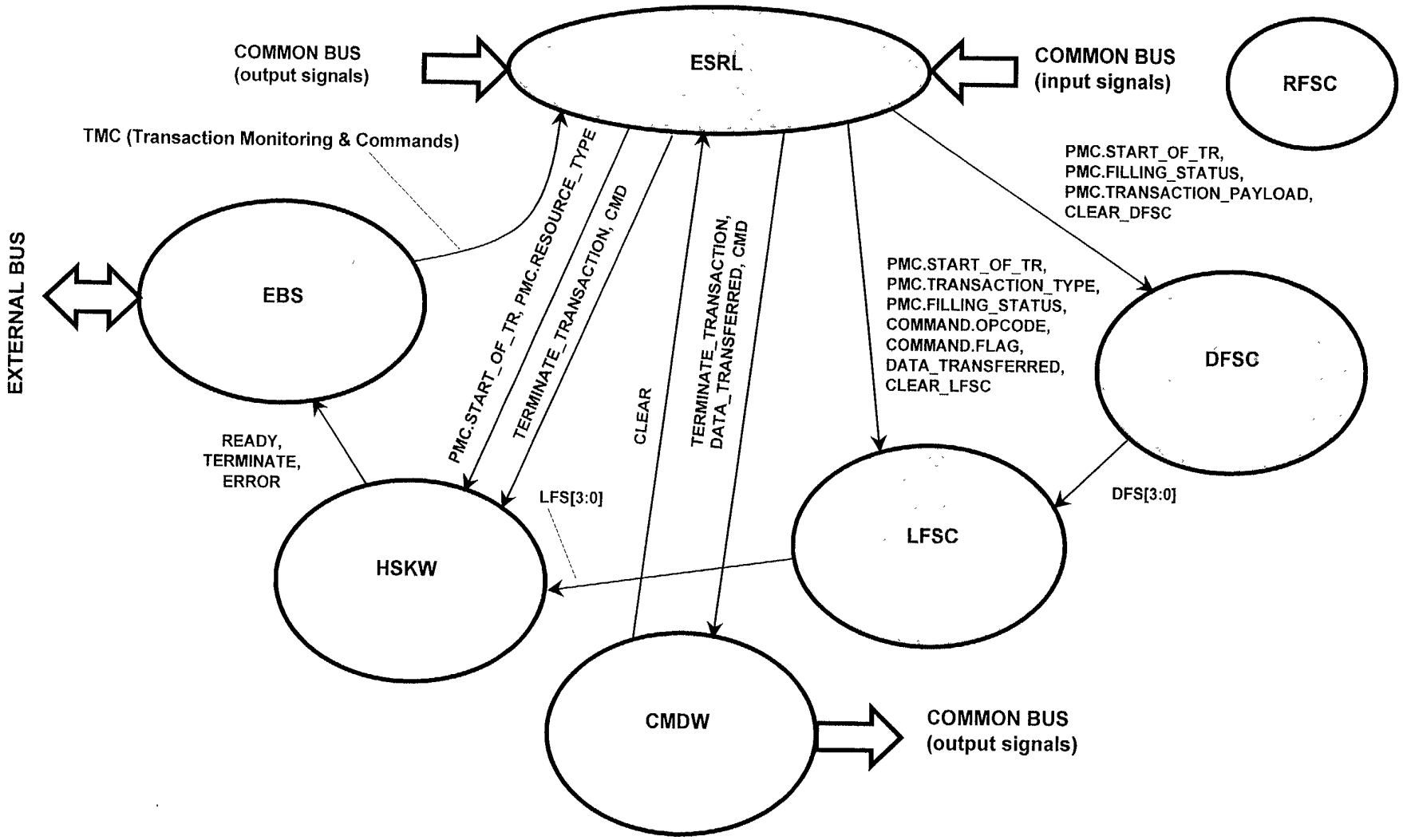


Figure 39

Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
WRITE Transaction to DMI PERIPHERAL1/Prefetchable resource
EXTERNAL BUS AGENT (master) Termination

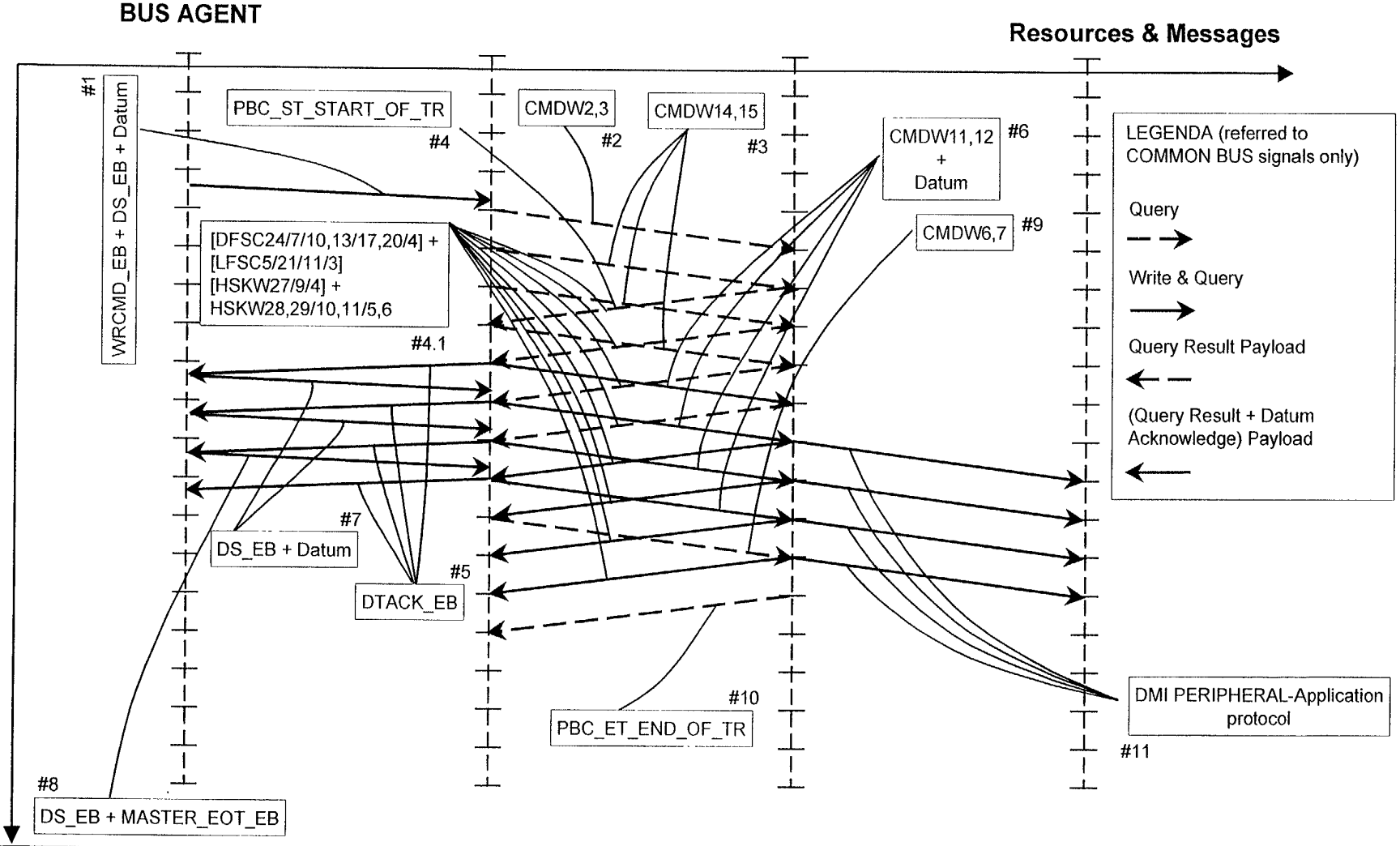


Figure 40

Message Sequence Chart
EXTERNAL BUS AGENT acting as Master - DMI acting as Slave
WRITE Transaction to DMI PERIPHERAL1/ Prefetchable resource
DMI (slave) Termination

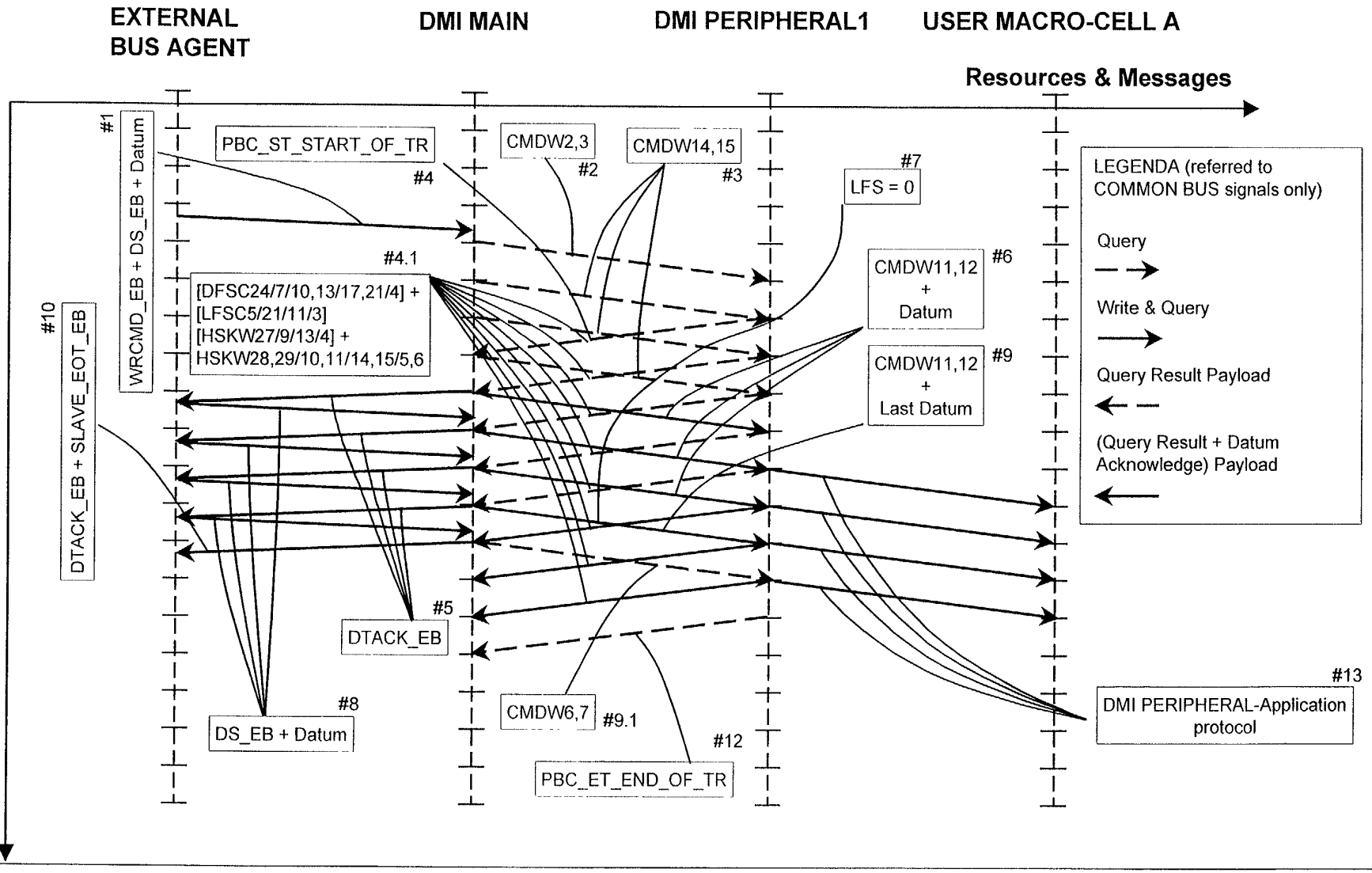


Figure 41

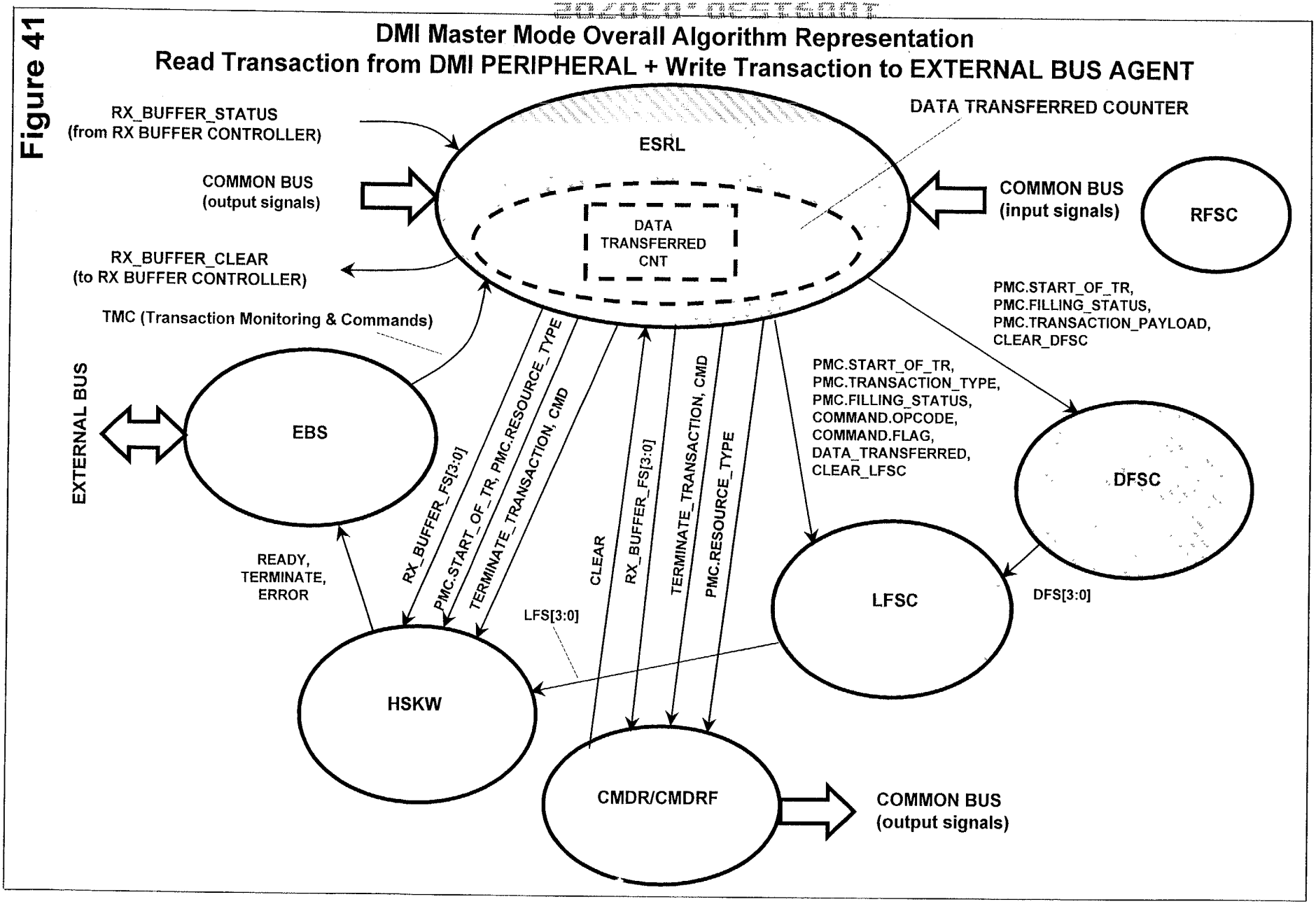

DMI Master Mode Overall Algorithm Representation
Read Transaction from DMI PERIPHERAL + Write Transaction to EXTERNAL BUS AGENT
RX_BUFFER_STATUS
(from RX BUFFER CONTROLLER)
COMMON BUS
(output signals)
RX_BUFFER_CLEAR
(to RX BUFFER CONTROLLER)
TMC (Transaction Monitoring & Commands)
EXTERNAL BUS
ESRL
DATA TRANSFERRED COUNTER
DATA
TRANSFERRED
CNT
COMMON BUS
(input signals)
RFSC
PMC.START_OF_TR,
PMC.FILLING_STATUS,
PMC.TRANSACTION_PAYLOAD,
CLEAR_DFSC
EBS
PMC.START_OF_TR,
PMC.TRANSACTION_TYPE,
PMC.FILLING_STATUS,
COMMAND.OPCODE,
COMMAND.FLAG,
DATA_TRANSFERRED,
CLEAR_LFSC
DFSC
RX_BUFFER_FS[3:0]
PMC.START_OF_TR, PMC.RESOURCE_TYPE
TERMINATE_TRANSACTION, CMD
CLEAR
RX_BUFFER_FS[3:0]
TERMINATE_TRANSACTION, CMD
PMC.RESOURCE_TYPE
READY,
TERMINATE,
ERROR
LFS[3:0]
LFSC
DFS[3:0]
HSKW
CMDR/CMDRF
COMMON BUS
(output signals)

Figure 42

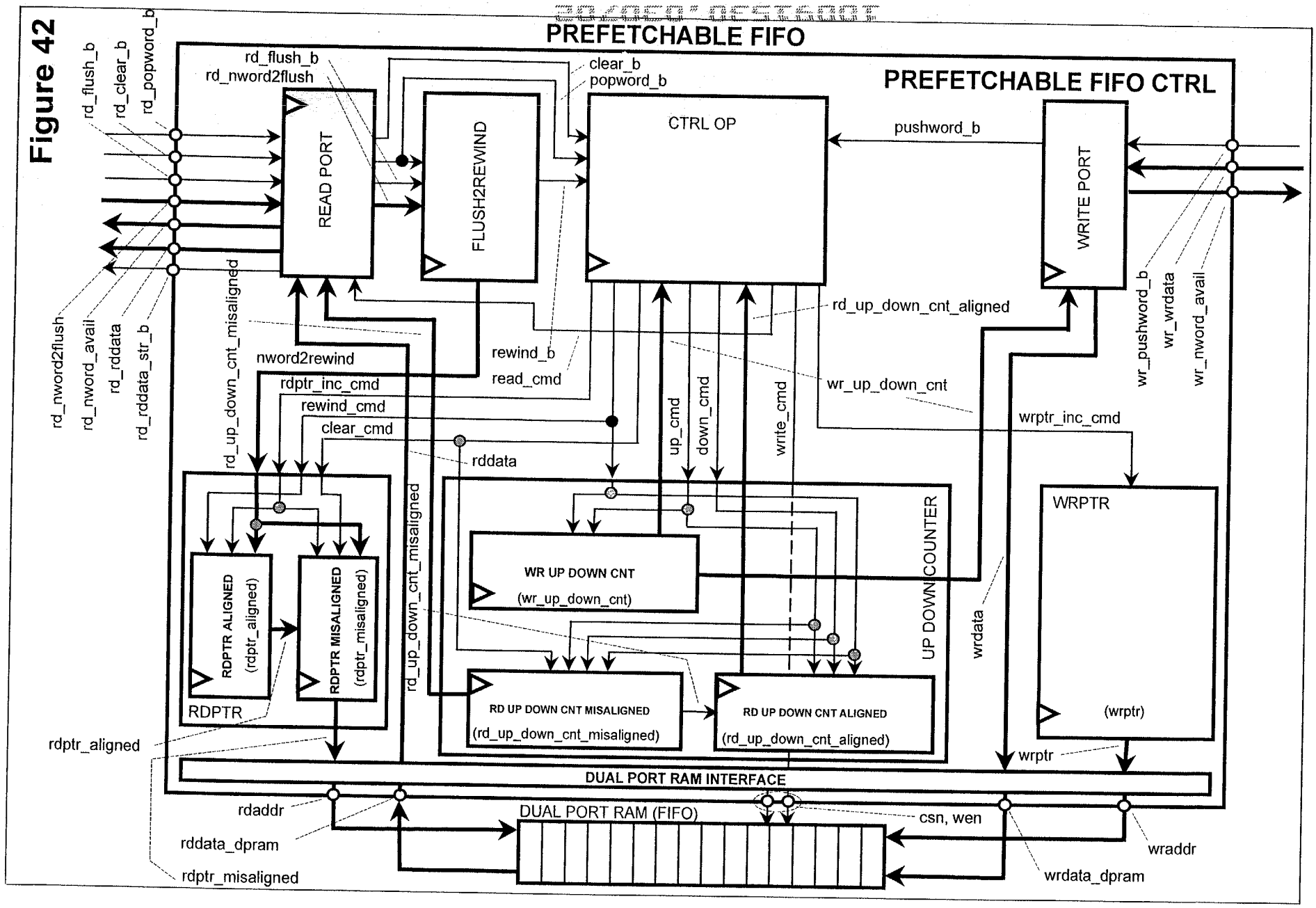

PREFETCHABLE FIFO
PREFETCHABLE FIFO CTRL
READ PORT
FLUSH2REWIND
CTRL OP
WRITE PORT
pushword_b
rd_flush_b
rd_nword2flush
clear_b
popword_b
rd_clear_b
rd_popword_b
rd_nword2flush
rd_nword_avail
rd_rddata
rd_rddata_str_b
rd_up_down_cnt_misaligned
nword2rewind
rdptr_inc_cmd
rewind_cmd
clear_cmd
rewind_b
read_cmd
rddata
rd_up_down_cnt_aligned
wr_up_down_cnt
wrptr_inc_cmd
up_cmd
down_cmd
write_cmd
wr_pushword_b
wr_wrdata
wr_nword_avail
RDPTR ALIGNED
(rdptr_aligned)
RDPTR MISALIGNED
(rdptr_misaligned)
RDPTR
WR UP DOWN CNT
(wr_up_down_cnt)
UP DOWN COUNTER
WRPTR
(wrptr)
wrdata
RD UP DOWN CNT MISALIGNED
(rd_up_down_cnt_misaligned)
RD UP DOWN CNT ALIGNED
(rd_up_down_cnt_aligned)
rdptr_aligned
wrptr
DUAL PORT RAM INTERFACE
rdaddr
DUAL PORT RAM (FIFO)
csn, wen
wraddr
rddata_dpram
wrdata_dpram
rdptr_misaligned

Figure 43

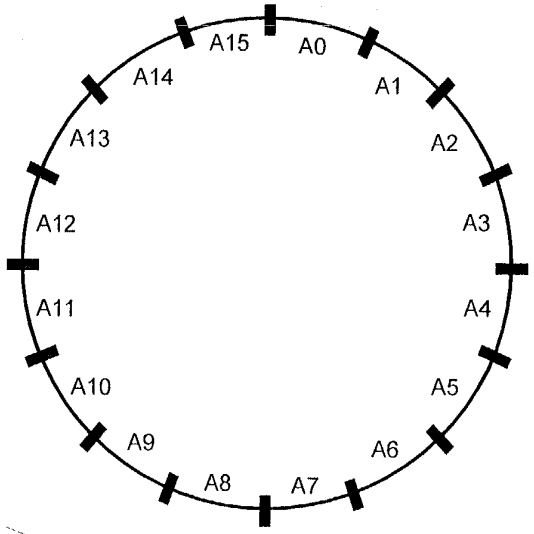
OVERALL STATUS

SNAPSHOT NUMBER = SN0;
BUS STATUS = IDLE;
FIFO FS = FIFO EMPTY;
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

-

PREFETCHABLE FIFO STATUS



UP DOWN COUNTER

wr_up_down_cnt = 0;
rd_up_down_cnt_misaligned = 0;
rd_up_down_cnt_aligned = 0;

RDPTR

rdptr_aligned = A0;
rdptr_misaligned = A0;

WRPTR

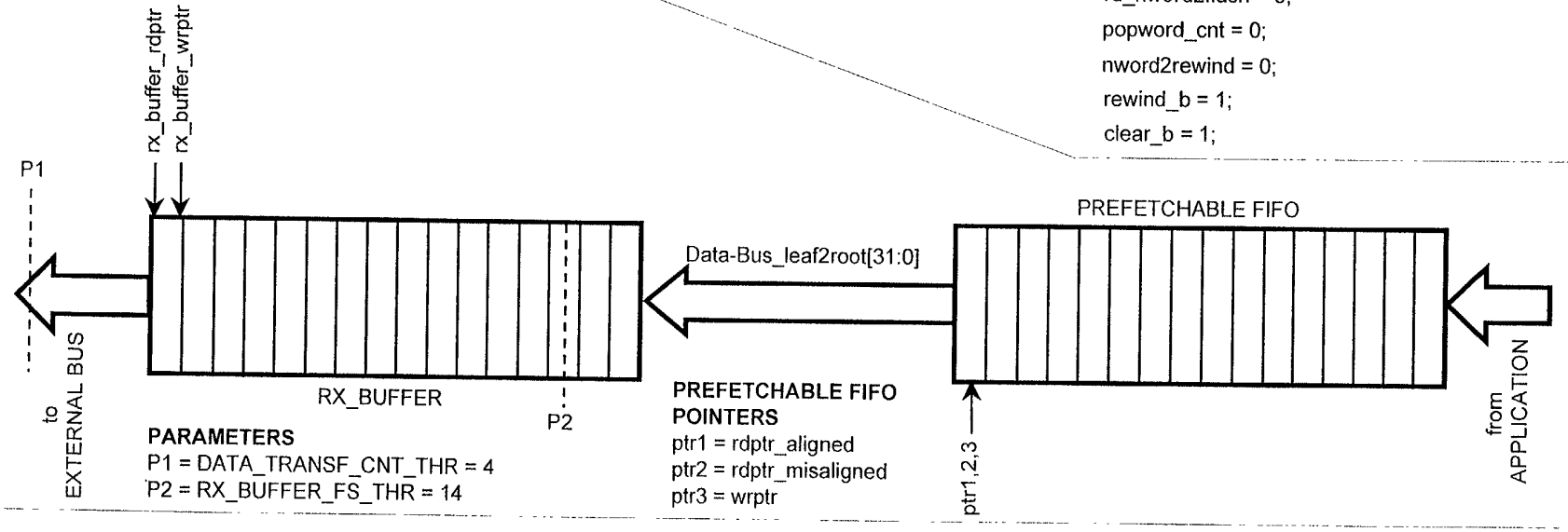
wrptr = A0;

FILLING_STATUS

rd_nword_avail = 0;
wr_nword_avail = 16;

FLUSH2REWIND

rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS

P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS

ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 44

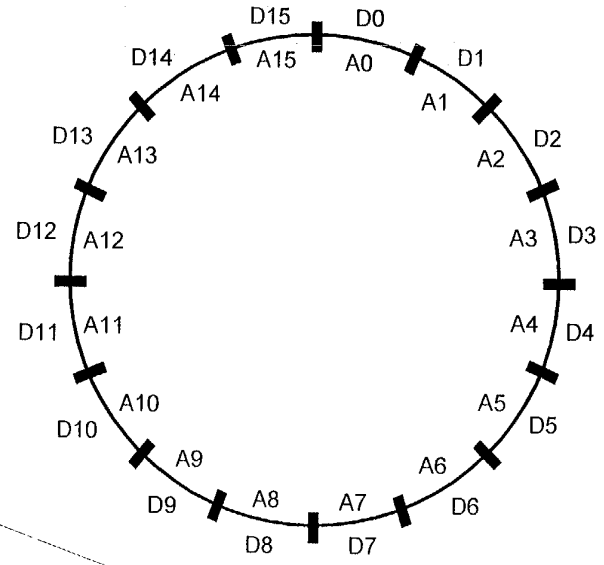
OVERALL STATUS

SNAPSHOT NUMBER = SN1;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

16 FIFO PUSH EXECUTED;

PREFETCHABLE FIFO STATUS



UP DOWN COUNTER

wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 16;
rd_up_down_cnt_aligned = 16;

RDPTR

rdptr_aligned = A0;
rdptr_misaligned = A0;

WRPTR

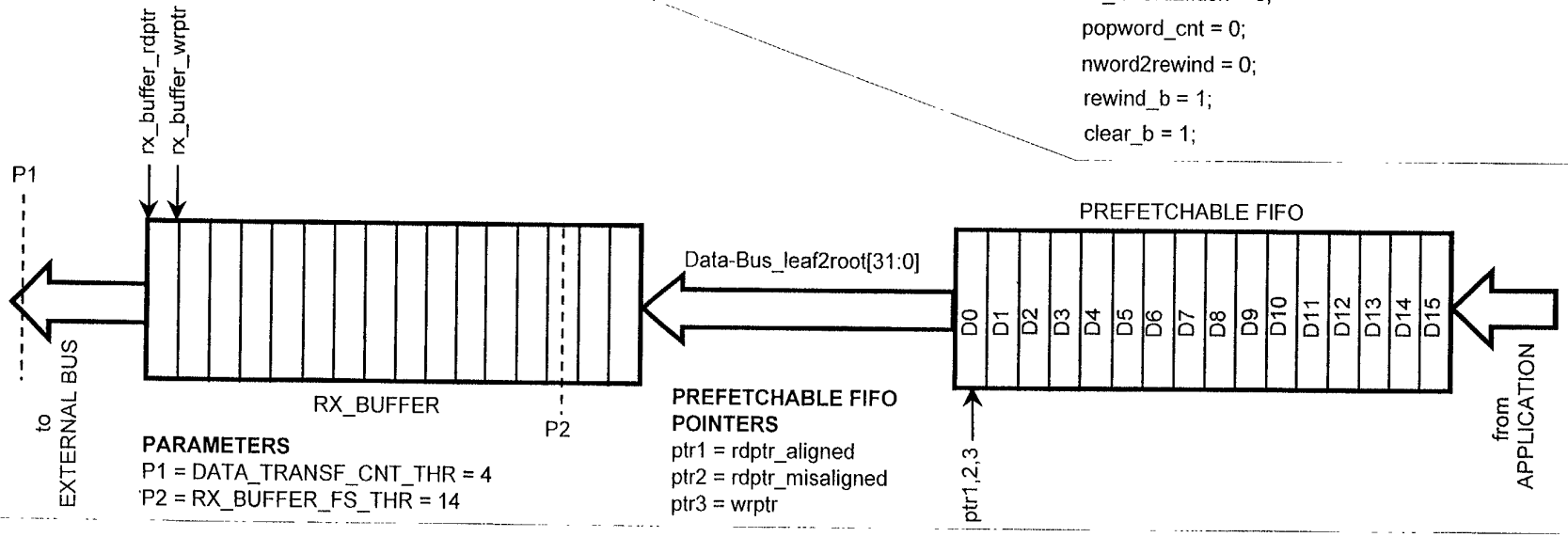
wrptr = A0
(one turn completed);

FILLING_STATUS

rd_nword_avail = 16;
wr_nword_avail = 0;

FLUSH2REWIND

rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS

P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS

ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 45

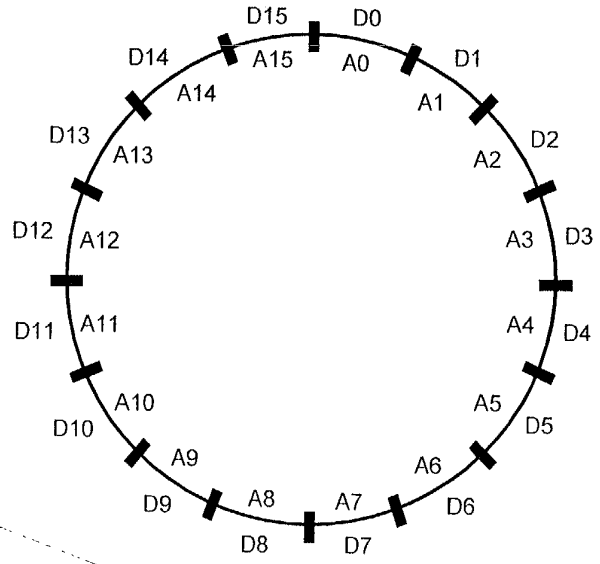
OVERALL STATUS

SNAPSHOT NUMBER = SN2;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

7 POP EXECUTED;

PREFETCHABLE FIFO STATUS



UP DOWN COUNTER

wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 9;

RDPTR

rdptr_aligned = A7;
rdptr_misaligned = A7;

WRPTR

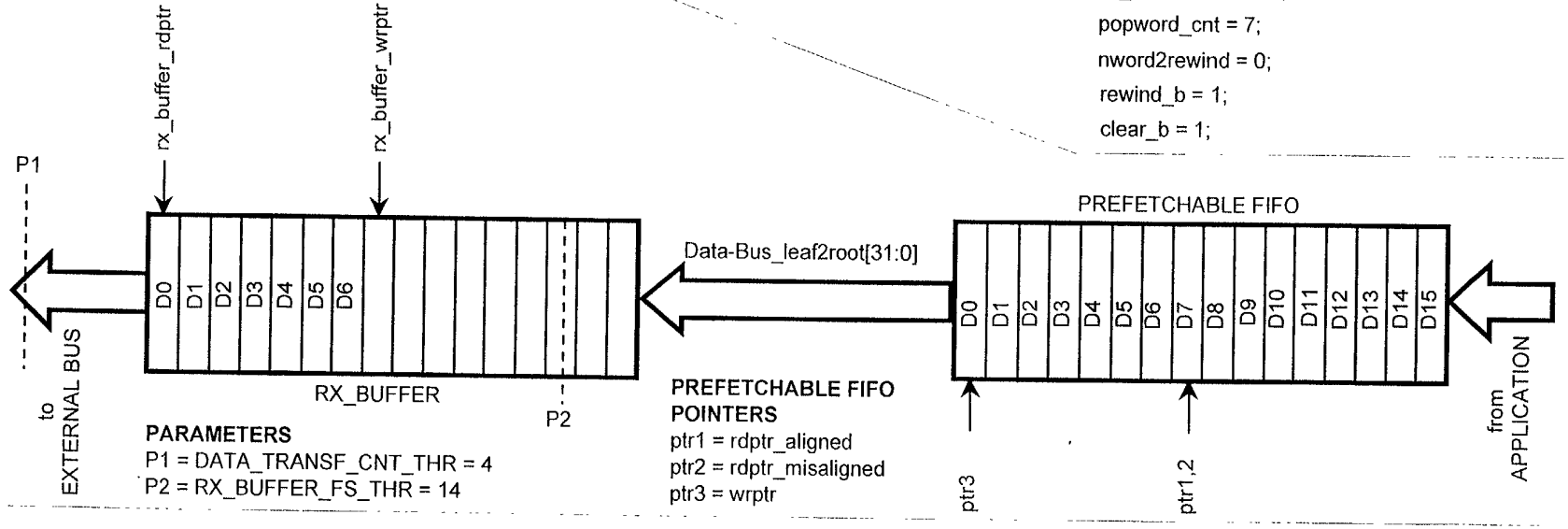
wrptr = A0;

FILLING_STATUS

rd_nword_avail = 9;
wr_nword_avail = 0,

FLUSH2REWIND

rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 7;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS

P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS

ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 46

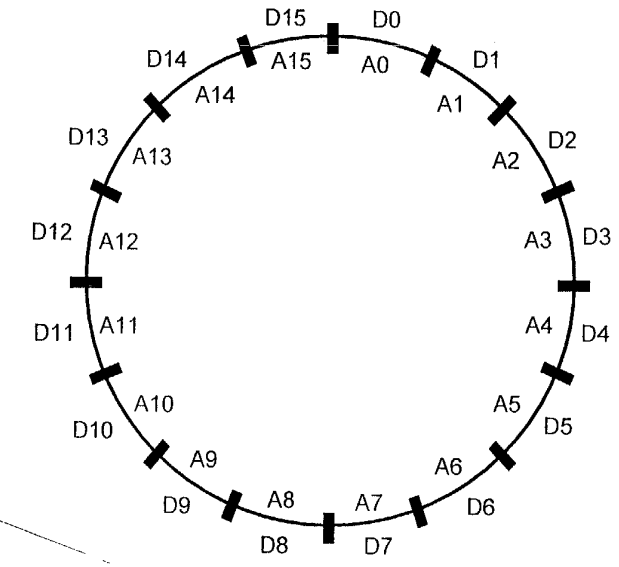
OVERALL STATUS

SNAPSHOT NUMBER = SN3;
BUS STATUS = WORKING;
FIFO FS = FIFO FULL;
DATA_TRANSFERRED_CNT = 4;

ISSUED / EXECUTED COMMANDS

FLUSH(4) ISSUED;

PREFETCHABLE FIFO STATUS



UP DOWN COUNTER

wr_up_down_cnt = 16;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 9;

RDPTR

rdptr_aligned = A7;
rdptr_misaligned = A7;

WRPTR

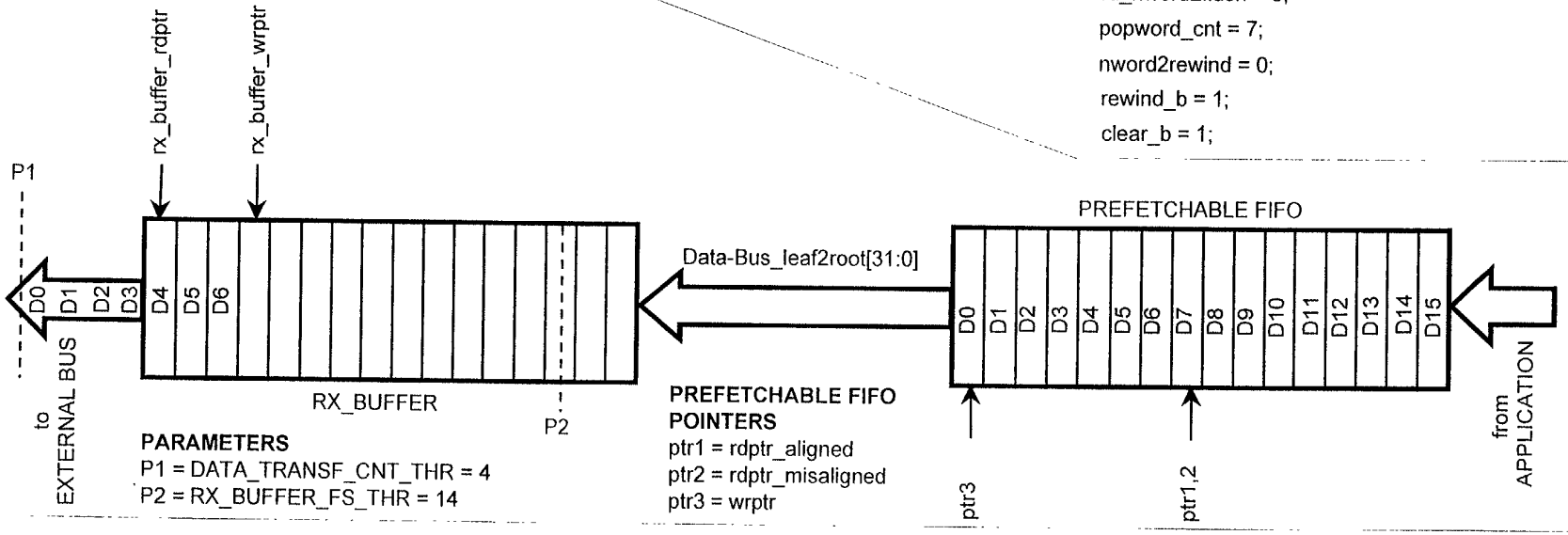
wrptr = A0;

FILLING_STATUS

rd_nword_avail = 9;
wr_nword_avail = 0;

FLUSH2REWIND

rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 7;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS

P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS

ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 47

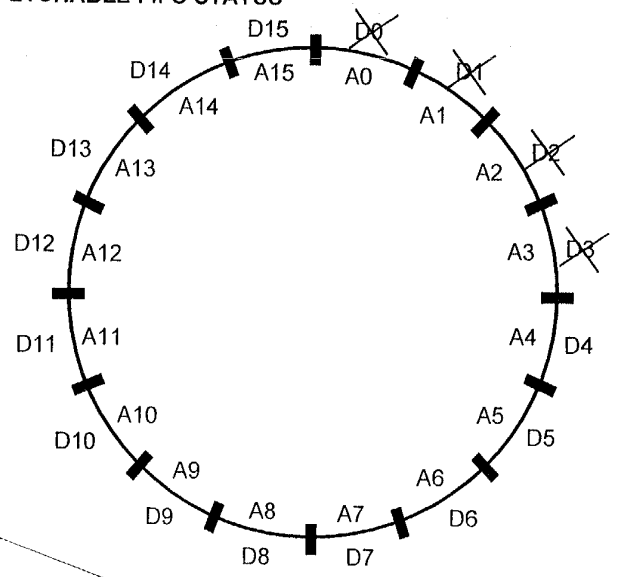
OVERALL STATUS

SNAPSHOT NUMBER = SN4;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

FLUSH(4) EXECUTED
7 POP EXECUTED;
REWIND(3) EXECUTED;

PREFETCHABLE FIFO STATUS



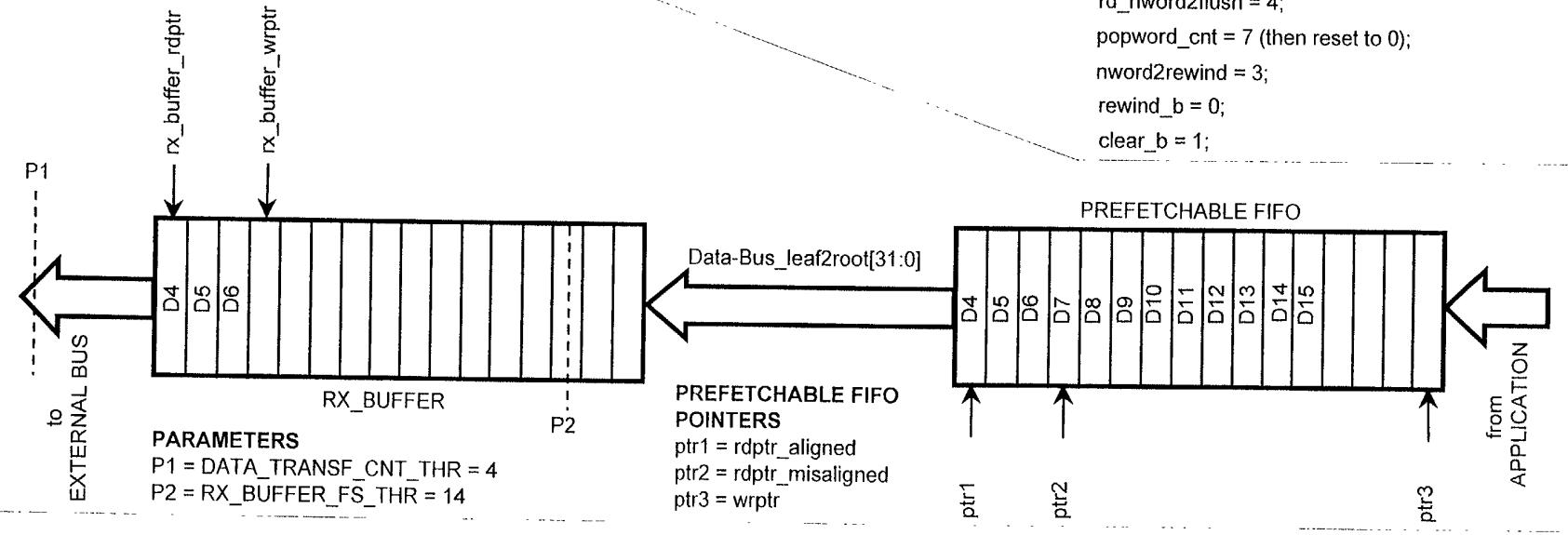
UP DOWN COUNTER
wr_up_down_cnt = 12;
rd_up_down_cnt_misaligned = 9;
rd_up_down_cnt_aligned = 12;

RDPTR
rdptr_aligned = A4;
rdptr_misaligned = A7;

WRPTR
wrptr = A0;

FILLING_STATUS
rd_nword_avail = 9;
wr_nword_avail = 4;

FLUSH2REWIND
rd_flush_b = 0;
rd_clear_b = 1;
rd_nword2flush = 4;
popword_cnt = 7 (then reset to 0);
nword2rewind = 3;
rewind_b = 0;
clear_b = 1;



PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 48

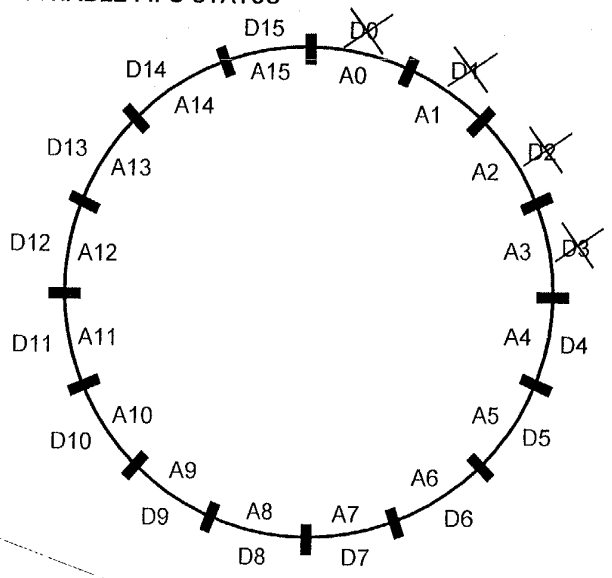
OVERALL STATUS

SNAPSHOT NUMBER = SN5;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

6 POP EXECUTED;

PREFETCHABLE FIFO STATUS



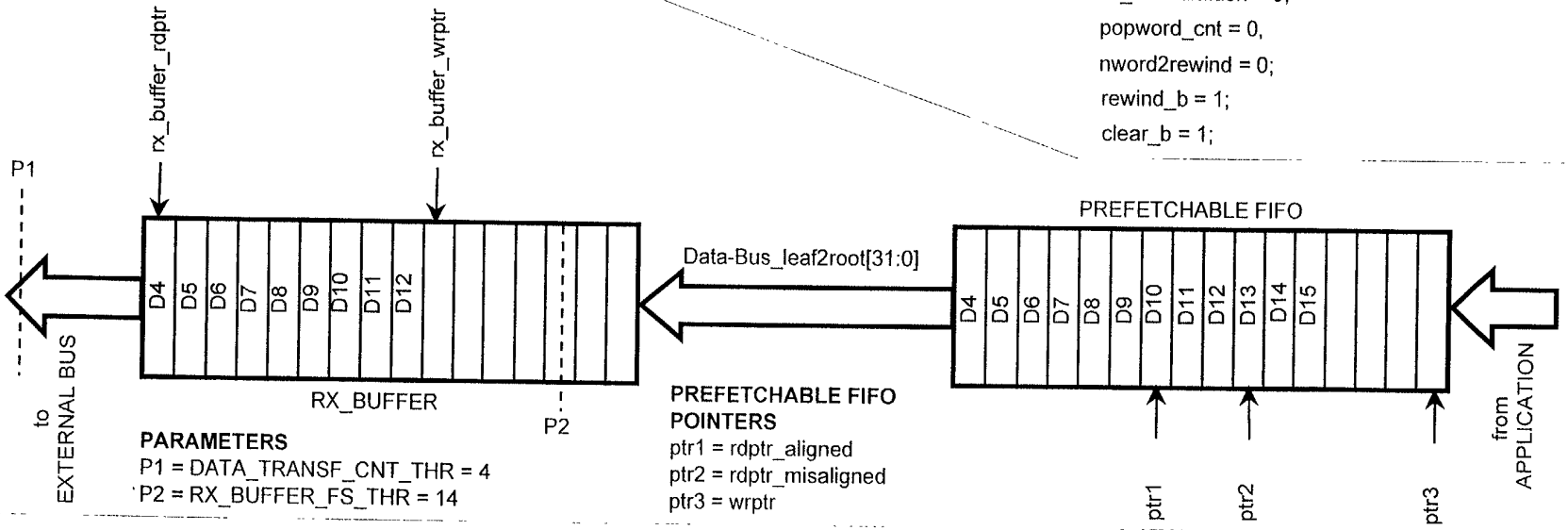
UP DOWN COUNTER
wr_up_down_cnt = 12;
rd_up_down_cnt_misaligned = 3;
rd_up_down_cnt_aligned = 6;

RDPTR
rdptr_aligned = A10;
rdptr_misaligned = A13;

WRPTR
wrptr = A0;

FILLING_STATUS
rd_nword_avail = 3;
wr_nword_avail = 4;

FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0,
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

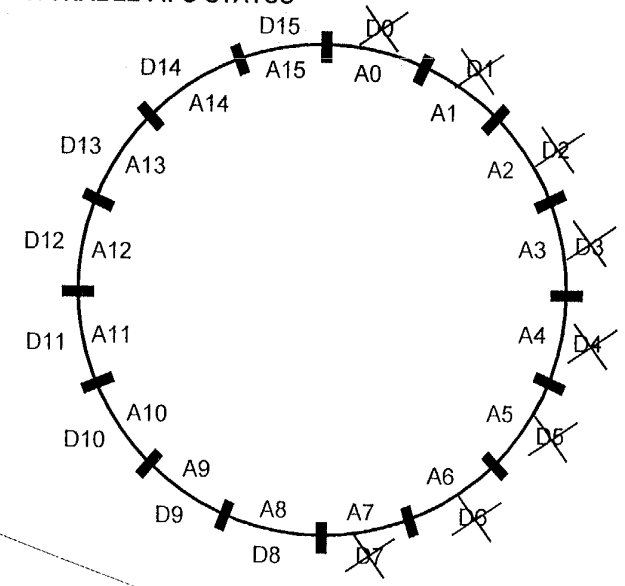
PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 49

OVERALL STATUS
SNAPSHOT NUMBER = SN6;
BUS STATUS = WORKING;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 4;

ISSUED / EXECUTED COMMANDS
CLEAR + FLUSH(4) EXECUTED:
6 POP EXECUTED;
REWIND(2) EXECUTED;

PREFETCHABLE FIFO STATUS



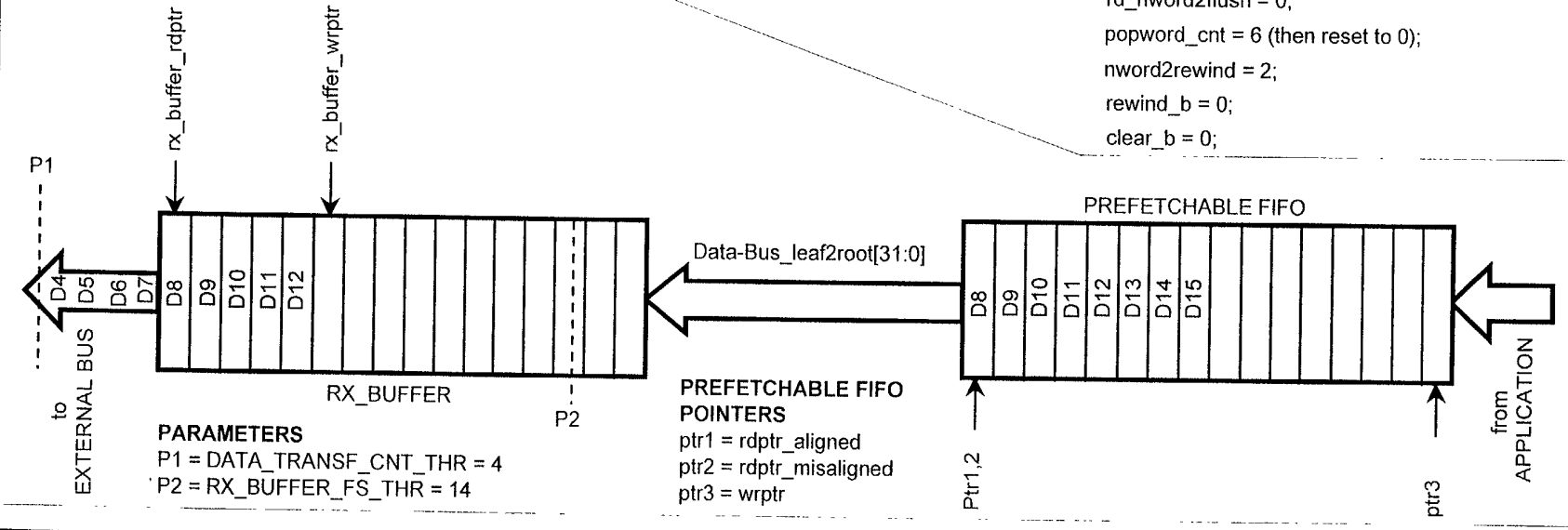
UP DOWN COUNTER
wr_up_down_cnt = 8;
rd_up_down_cnt_misaligned = 8;
rd_up_down_cnt_aligned = 8;

RDPTR
rdptr_aligned = A8;
rdptr_misaligned = A8;

WRPTR
wrptr = A0;

FILLING_STATUS
rd_nword_avail = 8;
wr_nword_avail = 8;

FLUSH2REWIND
rd_flush_b = 0;
rd_clear_b = 0;
rd_nword2flush = 0;
popword_cnt = 6 (then reset to 0);
nword2rewind = 2;
rewind_b = 0;
clear_b = 0;



PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 50

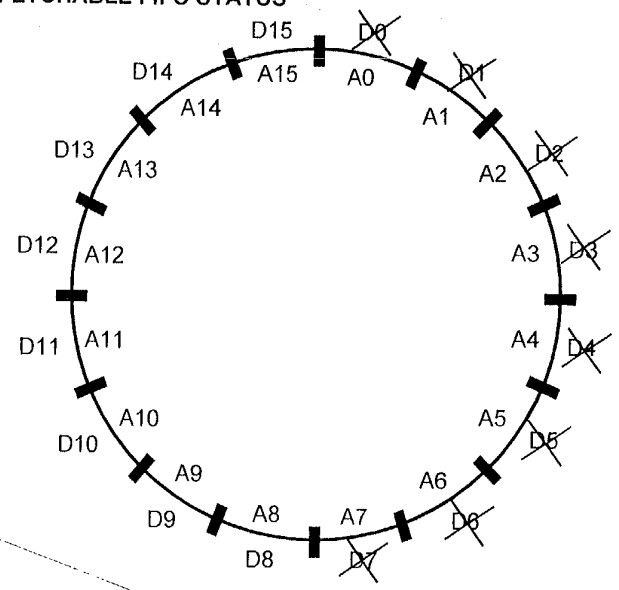
OVERALL STATUS

SNAPSHOT NUMBER = SN7;
BUS STATUS = IDLE;
FIFO FS = FIFO NOT FULL NOR EMPTY
DATA_TRANSFERRED_CNT = 0;

ISSUED / EXECUTED COMMANDS

RX_PIPELINE PURGE EXECUTED;

PREFETCHABLE FIFO STATUS



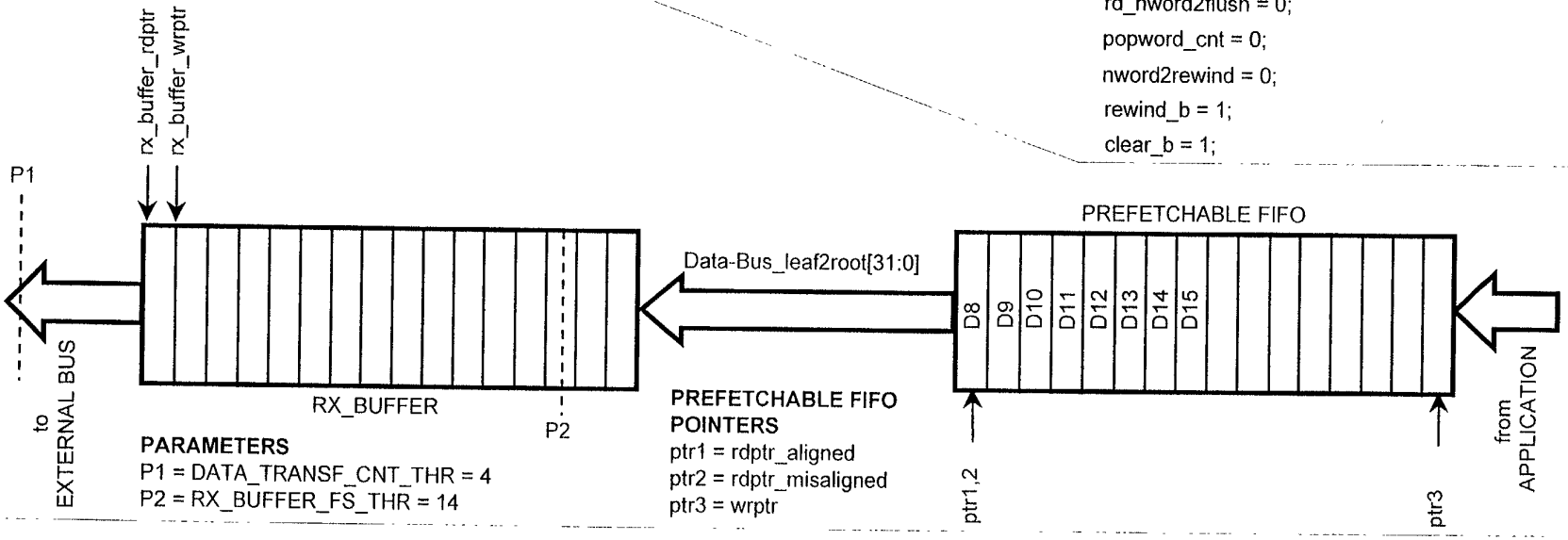
UP DOWN COUNTER
wr_up_down_cnt = 8;
rd_up_down_cnt_misaligned = 8;
rd_up_down_cnt_aligned = 8;

RDPTR
rdptr_aligned = A8;
rdptr_misaligned = A8;

WRPTR
wrptr = A0;

FILLING_STATUS
rd_nword_avail = 8;
wr_nword_avail = 8;

FLUSH2REWIND
rd_flush_b = 1;
rd_clear_b = 1;
rd_nword2flush = 0;
popword_cnt = 0;
nword2rewind = 0;
rewind_b = 1;
clear_b = 1;



PARAMETERS
P1 = DATA_TRANSF_CNT_THR = 4
P2 = RX_BUFFER_FS_THR = 14

PREFETCHABLE FIFO POINTERS
ptr1 = rdptr_aligned
ptr2 = rdptr_misaligned
ptr3 = wrptr

Figure 51

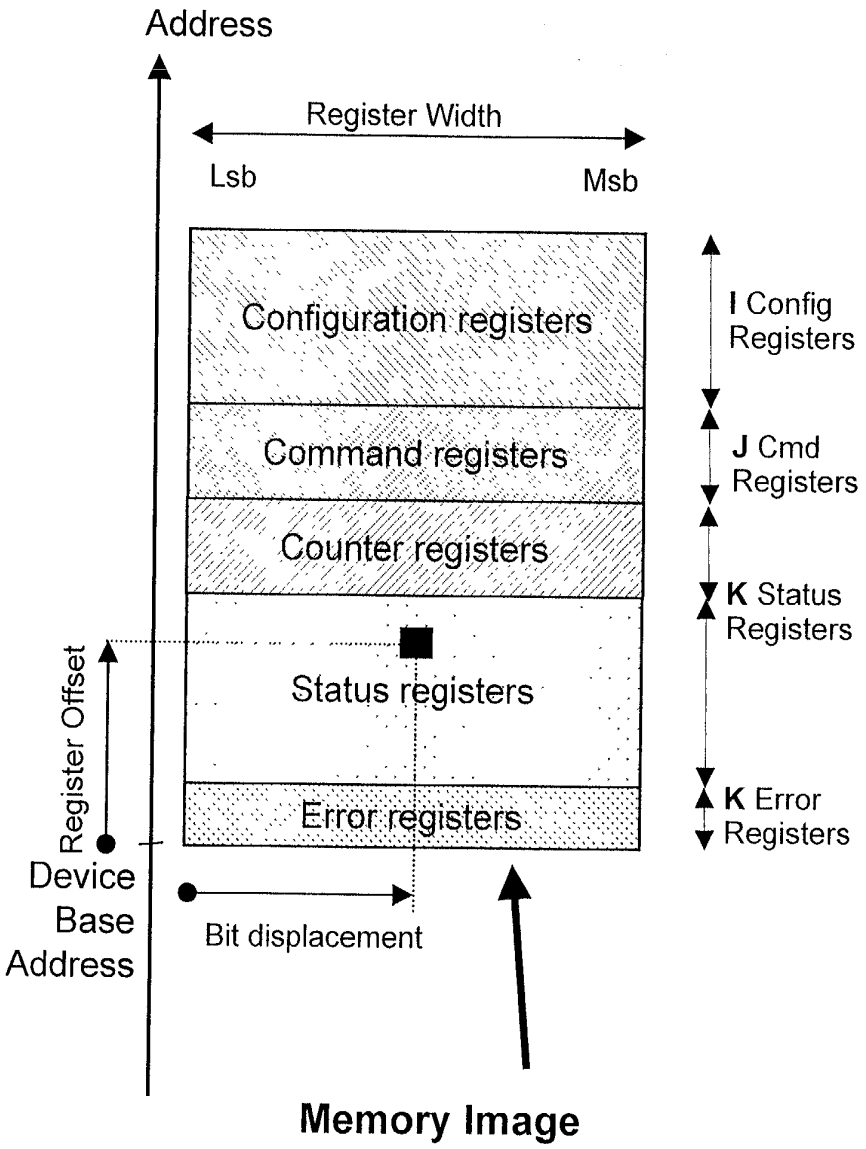

Address
Register Width
Lsb
Msb
Configuration registers
Command registers
Counter registers
Status registers
Error registers
I Config Registers
J Cmd Registers
K Status Registers
K Error Registers
Register Offset
Device Base Address
Bit displacement
Memory Image

Figure 52

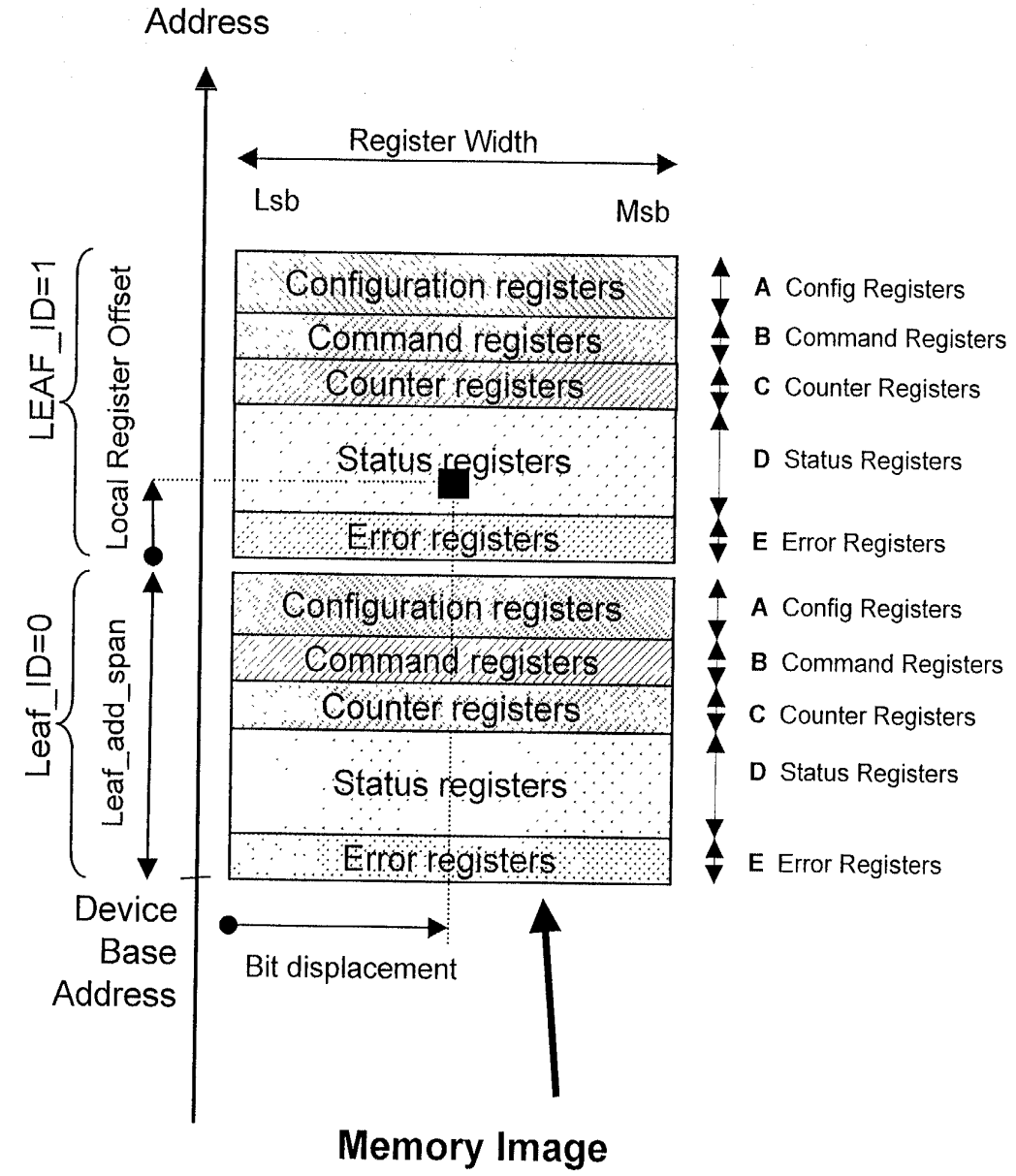

Address
Register Width
Lsb
Msb
LEAF_ID=1
Local Register Offset
Leaf_ID=0
Leaf_add_span
Configuration registers
Command registers
Counter registers
Status registers
Error registers
Configuration registers
Command registers
Counter registers
Status registers
Error registers
A Config Registers
B Command Registers
C Counter Registers
D Status Registers
E Error Registers
A Config Registers
B Command Registers
C Counter Registers
D Status Registers
E Error Registers
Device Base Address
Bit displacement
Memory Image

Figure 53

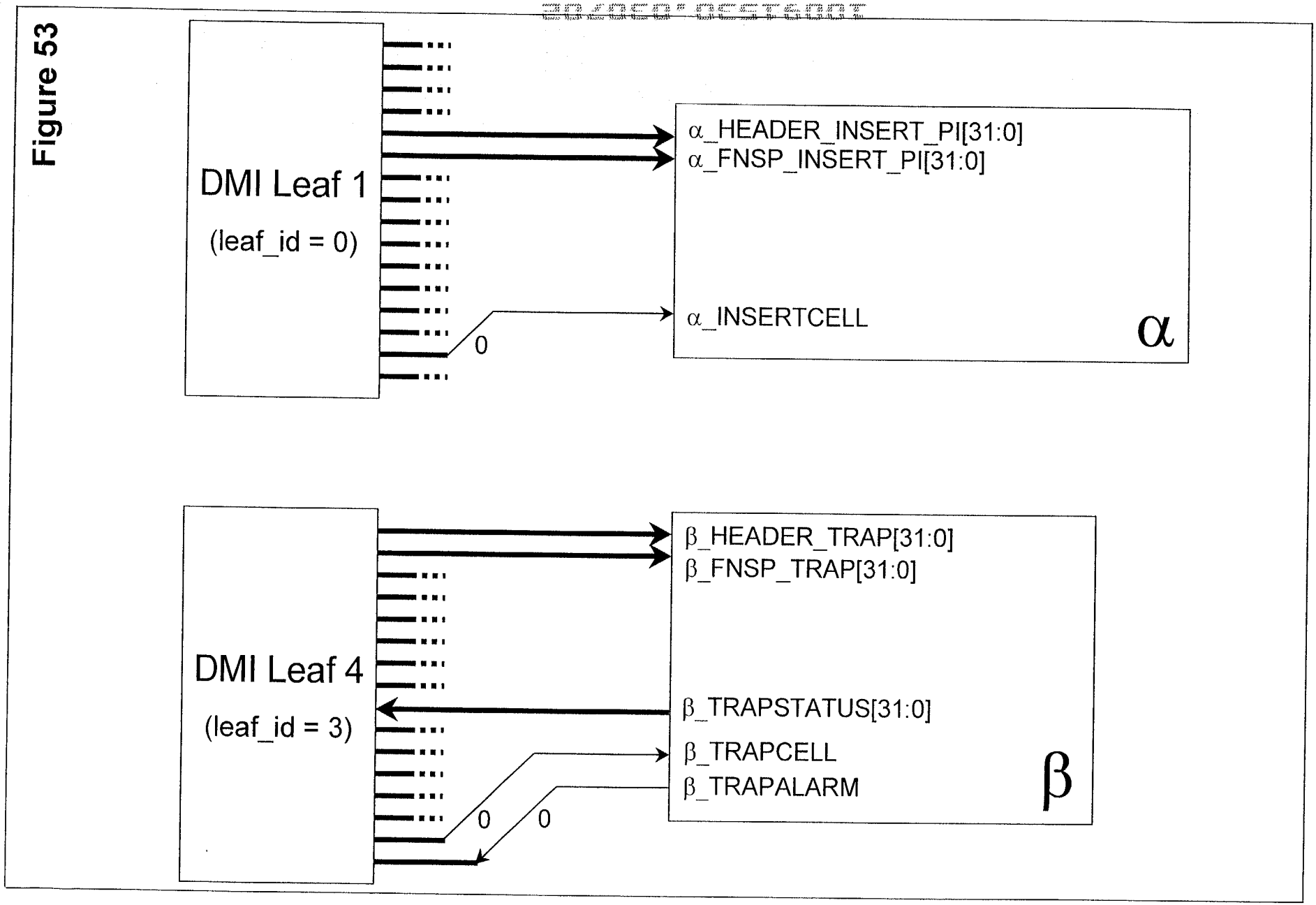

DMI Leaf 1
(leaf_id = 0)
α_HEADER_INSERT_PI[31:0]
α_FNSP_INSERT_PI[31:0]
α_INSERTCELL
0
α
DMI Leaf 4
(leaf_id = 3)
β_HEADER_TRAP[31:0]
β_FNSP_TRAP[31:0]
β_TRAPSTATUS[31:0]
β_TRAPCELL
β_TRAPALARM
0
0
β

Figure 54

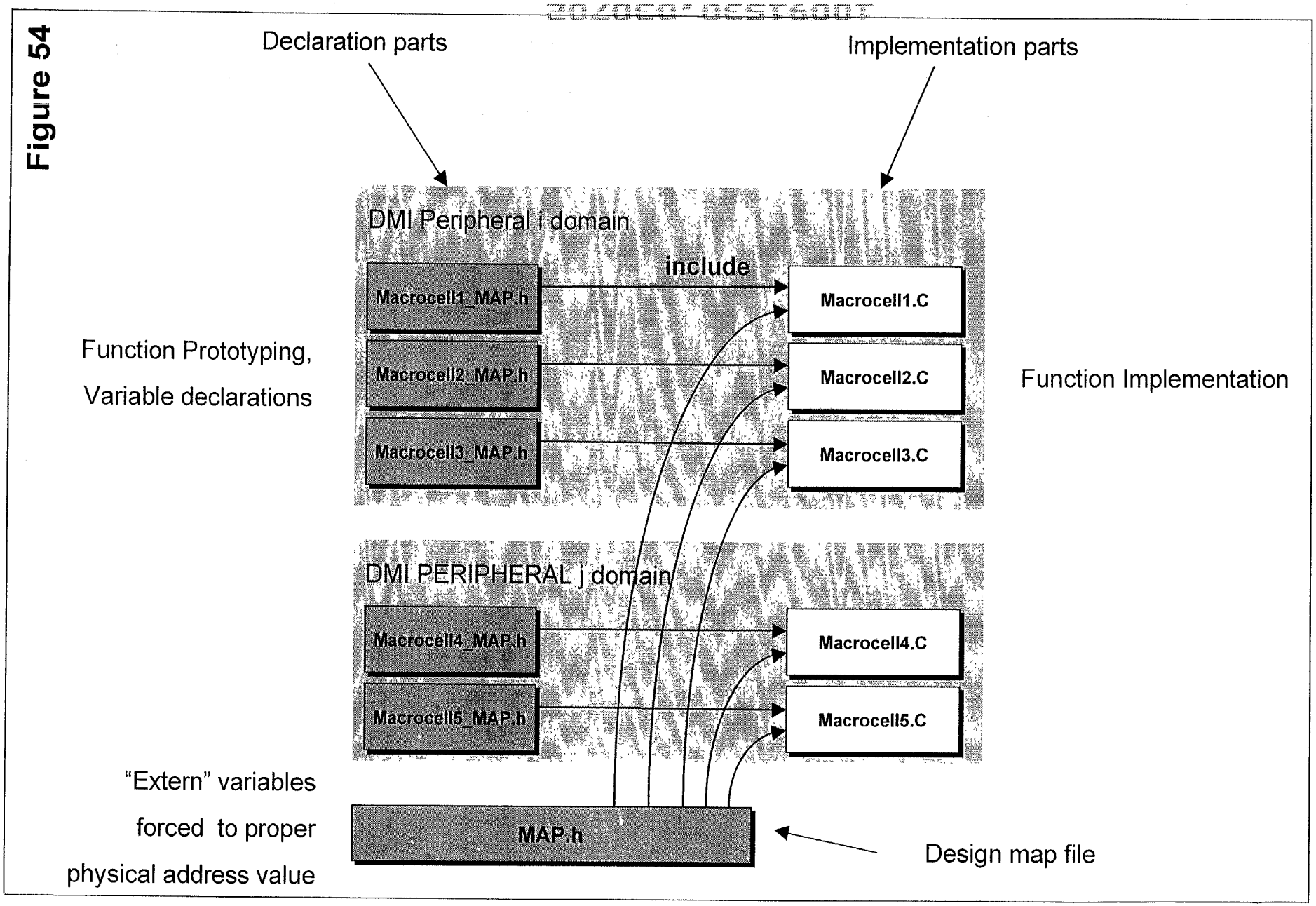
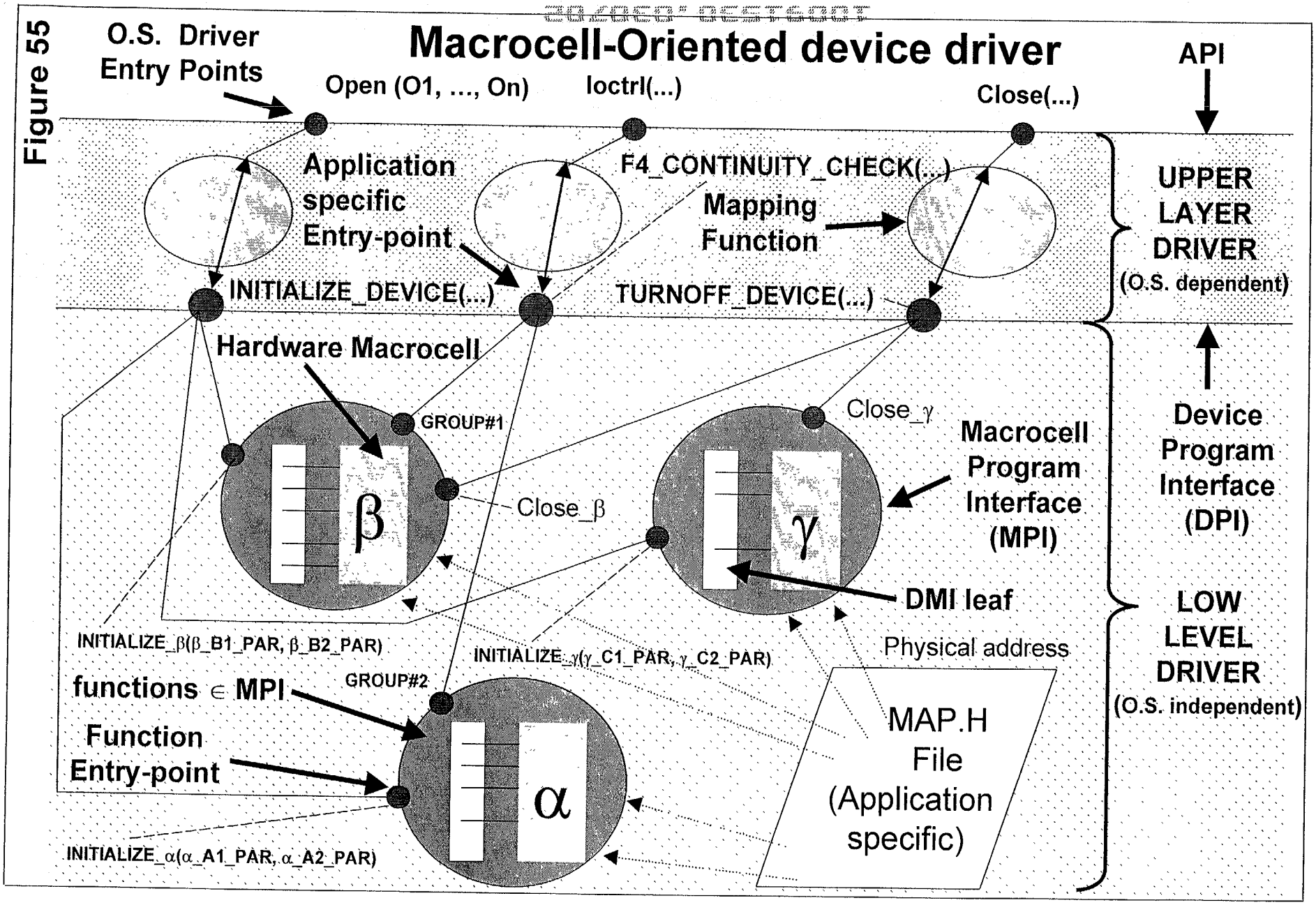

Declaration parts
Implementation parts
DMI Peripheral i domain
include
Macrocell1_MAP.h
Macrocell2_MAP.h
Macrocell3_MAP.h
Macrocell1.C
Macrocell2.C
Macrocell3.C
Function Prototyping,
Variable declarations
Function Implementation
DMI PERIPHERAL j domain
Macrocell4_MAP.h
Macrocell5_MAP.h
Macrocell4.C
Macrocell5.C
"Extern" variables
forced to proper
physical address value
MAP.h
Design map file

Figure 55


Macrocell-Oriented device driver
O.S. Driver Entry Points
Open (O1, ..., On)
Ioctrl(...)
Close(...)
API
Application specific Entry-point
F4_CONTINUITY_CHECK(...)
Mapping Function
UPPER LAYER DRIVER
(O.S. dependent)
INITIALIZE_DEVICE(...)
TURNOFF_DEVICE(...)
Hardware Macrocell
GROUP#1
Close_γ
Macrocell Program Interface (MPI)
Device Program Interface (DPI)
β
Close_β
γ
DMI leaf
INITIALIZE_β(β_B1_PAR, β_B2_PAR)
INITIALIZE_γ(γ_C1_PAR, γ_C2_PAR)
Physical address
functions ∈ MPI
GROUP#2
Function Entry-point
MAP.H File (Application specific)
LOW LEVEL DRIVER
(O.S. independent)
α
INITIALIZE_α(α_A1_PAR, α_A2_PAR)